System-AMG Approaches for Industrial Fully and Adaptive Implicit Oil Reservoir Simulations

I n a u g u r a l - D i s s e r t a t i o n

zur

Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität zu Köln

vorgelegt von

Sebastian Gries

aus Siegburg

Köln

2016

Berichterstatter:	Prof. Dr. Ulrich Trottenberg
(Gutachter)	Prof. Dr. Axel Klawonn
	Prof. Hamdi Tchelepi, Ph.D. (Stanford University)

Tag der mündlichen Prüfung: 20.01.2016

Abstract

A continuous increase of sub-surface flow models in size and physical complexity makes efficient and reliable linear solution approaches crucial for successfully applying reservoir simulations. Due to the essential objective of simulating diffusive fluid flux, algebraic multigrid methods are a natural option to consider. However, the application may not be straight-forward, as the solver has to cope with linear systems that are influenced by various physical effects.

In this thesis we will discuss AMG-based solution approaches for Black-Oil and compositional models for fluid flow, as well as for models that additionally take thermal and mechanical effects into account. We will discuss the properties of the matrices that describe the linear systems and we will see the impact of different simulated effects.

As Black-Oil models form the basis also for more sophisticated models, we will discuss a robust System-AMG approach for these simulations first. This will include preparatory matrix transformations that aim at ensuring the applicability of AMG. With this approach, we will be able to solve highly challenging problems from industrial simulations robustly and efficiently. We will then extent this approach to compositional, thermal and geomechanical problems.

Finally, we will discuss some aspects of further improving the performance of System-AMG. This will involve algorithmic modifications that give AMG approaches with better computational efficiency, but we will also discuss some implementational aspects, e.g. regarding concurrency of incomplete factorizations.

Zusammenfassung

Die kontinuierliche Zunahme von Problemgröße und physikalischer Komplexität in Modellen für Fluss durch poröse Medien lässt linearen Löseransätzen, die zugleich effizient und verlässlich sind, eine entscheidende Bedeutung für erfolgreiche Reservoir-Simulationen zukommen. Da es im Kern um die Simulation diffusiven Flusses geht, bieten sich algebraiche Mehrgittermethoden natürlicherweise an. Allerdings ist ihre direkte Anwendung unter Umständen schwierig, da die zu lösenden Systeme von zahlreichen physikalischen Effekten beeinflusst werden.

In dieser Dissertation werden sowohl reine Flussmodelle für Mehrphasen- und Mehrkomponentenfluss behandelt, als auch Modelle die zusätzlich thermische und mechanische Einflüsse berücksichtigen.

Es werden zunächst die Eigenschaften der Matrizen betrachtet, die die zu lösenden Systeme beschreiben. Hierbei werden die Einflüsse der verschiedenen physikalischen Effekte herausgestellt.

Dann wird ein System-AMG Ansatz für Schwarzöl-Simulationen beschrieben, da diese Modelle die Basis für komplexere Simulationen bilden. Dieser Ansatz beinhaltet einen Vorbereitungsschritt für die Matrix, der für eine robuste Anwendbarkeit von AMG sorgt. Auf diese Weise lassen sich auch sehr komplizierte lineare Systeme aus industriellen Simulationen erfolgreich und effizient lösen. Im Anschluss wird beschrieben wie sich dieser Ansatz auf Mehrkomponentenprobleme sowie auf Simulationen, die thermale und mechanische Effekte berücksichtigen, erweitern lässt.

Zum Abschluss werden einige Aspekte zur Laufzeitverbesserung behandelt. Dies betrifft auf der einen Seite algorithmische Änderungen am Löseransatz. Zum andern wird auch die effiziente Realisierung, unter anderem Parallelisierungsaspekte unvollständiger Matrixzerlegungen, vorgestellt.

Contents

List of Figures

List of Tables

Nomenclature

In this nomenclature we only list symbols and notations that are used throughout this thesis. In some sections we may locally define some further symbols. Such symbols are not listed here.
In order to follow the general notations from different disciplines (e.g., modeling of sub-surface flow and multigrid), some symbols may be used with a different meaning in different chapters. However, which meaning is intended will always be clear from the context.

Latin Letters

A, B, C, D	Square matrices
A_{XY}	Sub-matrix describing relations between physical unknowns X and Y
C_L, C_R	Left- and right-scaling, respectively
E	Young's modulus
$[E]$	Discrete energy balance equation
$[E^{Accu}]$	Part of $[E]$ that results from accumulation (analogously with $FluidFlux$, $HeatFlux$ and $Source$)
G_A	Adjacency graph of matrix A
GL_A	Graph Laplacian of matrix A
G_{str}	Graph of strong couplings
H	Depending on the context: Thermal enthalpy or grid size
J	Jacobian

$\vec{\vec{K}}, K$	Permeability
$K_{h,H}$	Coarse grid correction operator
$\vec{\vec{K}}_T, K_T$	Thermal permeability
M	Iteration operator
$[M_\alpha]$	Discrete mass balance equation of phase α
$[M_\alpha^{Accu}]$	Part of $[M_\alpha]$ that results from accumulation (analogously with $Flux$ and $Source$)
M_{Biot}	Biot's modulus
M_c^{mol}	Molar mass of component c
N_α,N_c	Number of moles in phase α or component c, respectively
P	Primary matrix
R_{og}	Oil-gas ratio (amount of gas that is dissolved in the oil phase)
S_α	Saturation of phase α
S	Depending on the context: Saturations or smoothing operator
T	Temperature
T_α	Transmissibility
U	Internal energy
V	Volume
WI	Well index
$X_{c,\alpha}$	Concentration of component c in phase α
d	Depth
e	Error vector
f	Right-hand-side
g	Gravitational constant

h	Grid size with $h < H$
k_{rel}	Relative permeability
m	Mass
p_α	Pressure of phase α
p_{bp}	Bubble point pressure
$p_{\alpha\beta}^{cap}$	Capillary pressure between phases α and β
q_α	Source term for phase α
r	Residual
t	Time
u	Displacement
$\vec{v}$	Velocity
x, y	Solution vector

Greek Letters

α, β, γ	Phases
α_{Biot}	Biot's effective stress coefficient
δ_i^j	DRS weight for unknown j in cell i
ϵ	Depending on the context: Strain or anisotropy
ϵ_v	Volumetric strain
λ	Depending on the context: Mobility or reduction factor
ρ	Depending on the context: Density or spectral radius
μ	Viscosity

ν	Poisson's ratio
φ	Porosity
σ	Depending on the context: Stress or smoothing factor
σ'	Effective stress

Subscripts

c	Component
tot	Total value (i.e., weighted average over all considered fluids; see Equation (2.4))
α, β, γ	Fluid phases

General Notation

$[\mathcal{X}]_i^n$	Property (or equation) $\mathcal{X}$ in grid cell i at time step n
$\tilde{\mathcal{A}}$	Matrix $\mathcal{A}$ is scaled in some way
$\mathbb{1}$	Identity matrix

Chapter 1

Introduction

The numerical simulation of sub-surface flow processes plays a key role in the design of resource recovery in the oil and gas industry. Such simulations always require solving linear systems of equations in the numerical kernel. However, the complex physics of multiphase sub-surface flow results in linear systems that are challenging for linear solvers. This especially holds for iterative ones that, due to the problem sizes, are the methods of choice. The simulation of *enhanced oil recovery* (EOR) techniques that involve thermal influences and/or geomechanical forces, adds additional complexities and degrees of freedom to the models and the resulting linear problems. Moreover, the continuous increase in model resolutions leads to corresponding growth in problem sizes. Both the increasing complexity and size of the systems makes their solution by far the most time consuming part of today's reservoir simulators.
The application of the underlying simulation techniques is not limited to the exploitation of oil and gas reservoirs. They can also be applied to further sub-surface flow problems, including gas storage facilities, the sequestration of carbon dioxide and nuclear waste deposits. Especially the latter are of increasing importance. There are also applications beyond sub-surface flow. For instance, the simulation of bone marrow in medical engineering can be achieved using the same concepts of porous media flow.
For all these simulation applications, there is a need for efficient linear solution approaches that are robust and reliable.

In this thesis we will discuss the application of efficient linear solver methods for

various types of porous media flow simulations that involve multiple kinds of fluids and various chemical components. This will also include the application for coupled linear systems that involve thermal and mechanical effects. We will see that so-called System-AMG provides a framework for constructing efficient and robust linear solver approaches in the outlined types of simulations.
Our discussion will focus on reservoir simulations, as research on simulation of porous-media flow is mainly driven by this application. Reservoir simulations play an important role in the oil industry to optimize the exploitation strategy for hydrocarbon reservoirs. As the costs of complex well-bores can easily reach tens of millions of dollars, each of them should result in as much oil or gas being produced as possible. In conventional discovery processes, a combination of production and injection wells is used. A fluid is injected into the reservoir, in order to increase the in-situ pressure and push the oil to a production well. This injected fluid is classically water, but also the use of gas becomes increasingly popular. In particular carbon dioxide should be mentioned, as it is at the same time sequestrated this way.
The complex structure of the sub-surface rock causes a high sensitivity of a well's efficiency on its exact location. The reservoir is not a large cavity, but rather a sponge-like structure with various interconnected pores, which yields highly heterogeneous permeability fields. The oil is located within these pores and even with the utilization of injections, typically only up to a quarter of the oil within a reservoir can be recovered efficiently. Highly viscous hydrocarbons then still remain within the pore structure. Moreover, there might be substantial parts of the reservoir with only a small, or even no transmissibility to its main parts. This especially applies to shale reservoirs. In order to also produce this remaining oil, EOR techniques are used. These techniques essentially employ heat, force and chemistry:

- The oil's viscosity can be reduced by increasing the temperature within the reservoir - either by injecting steam, or by in-situ heating facilities and even combustion.
- High pressures can be used to induce mechanical forces and widen the pores of the reservoir field. This technique can be extended in order to grow existing, or create new fractures (so-called fracking).
- Certain chemicals and foams can be injected into the reservoir in order to detach oil from pores.

We are concerned with different types of fluid in the flow process and may also consider further physical effects. Hence, the resulting linear systems involve several different types of corresponding physical unknowns. In order to overcome strict limitations for the time step sizes, at least most of them are treated implicitly in the discretization, which leads to the *fully and adaptive implicit* (FIM and AIM) simulation models that we will be concerned with in this thesis. The differential equations in the model formulations are highly non-linear and the linear systems that we intend to solve in fact are described by Jacobian matrices that are employed in the linearization process, for instance by Newton's method.
All the physical processes outlined above are handled by today's reservoir simulators, which makes these linear systems rather challenging for solvers. Even in the simulation of reservoirs in their initial state without well-bores, the systems involve different types of unknowns with different physical backgrounds and are characterized by strong heterogeneities. The consideration of well-bores results in additional impacts on the linear system side.

As the objective of reservoir simulations essentially is the simulation of diffusive fluid flow, multigrid methods are a natural option to consider. For discretized partial differential equations (PDEs) describing "diffusion-driven" processes, multigrid methods have proven to provide efficient solvers, as discussed by Brandt and Livne or Trottenberg et.al. [14, 97]. These methods exploit a hierarchy of grids at different resolutions in order to uniformly reduce all error components[1] of a solution iterate. This property makes multigrid methods superior to non-hierarchical, i.e. one-level, solution methods like relaxations or incomplete factorizations.
One way of utilizing a grid hierarchy is to discretize the initial PDE on grids of different resolutions and then construct geometry-based transfer operators between them. This is the approach of *geometric multigrid* (GMG), which is highly efficient where applicable. However, analog grids of different mesh sizes, together with reasonable transfer operators, may be extremely difficult to find, if not even practically impossible, in the case of complex unstructured grids featuring strong heterogeneities.
A way to overcome this limitation is provided by the idea of *algebraic multigrid* (AMG) methods, as described by Ruge and Stüben [85, 95]. This approach also exploits a hierarchy of grids, or rather levels, to uniformly reduce the different components of an error function. However, this hierarchy is constructed automatically by the method. Given that the linear system to be solved results from the

[1] The components of an error function are defined in terms of its Fourier decomposition.

discretization of an elliptic PDE, the matrix exhibits certain properties that AMG can exploit in order to construct hierarchy and transfer operators. In this way, the levels are directly adapted to a given initial matrix and, hence, the corresponding grid and varying material parameters.

A state-of-the-art solution approach for the linearized systems from reservoir simulations is the *Constrained Pressure Residual* (CPR) method, proposed by Wallis, et.al. [101, 102]. This method can be regarded as a kind of two-stage preconditioning. As the fluid flow is driven by pressure differences, the sub-system that is affiliated with the pressure unknown is expected to describe the diffusive part of the entire system. This sub-system is approximately decoupled from the other physical unknowns by certain matrix transformations. In each iteration of the CPR method then a rough pressure approximation is computed as a first step, which serves as an initial guess for an incomplete factorization iteration in the second step.
Due to its known efficiency in solving linear problems resulting from diffusive processes, algebraic multigrid is already a popular choice for computing the pressure approximation in CPR's first step, yielding a CPR-AMG method. Unfortunately, while working efficiently in many simulations, a failure of the CPR-AMG method is observed in a significant number of simulations in practice. This failure nearly always is caused by AMG-related issues in the computation of the pressure approximation. Moreover, whether or not CPR-AMG fails depends on the way the pressure sub-problem is extracted from the full system. This is because the transformation that is used for the approximate decoupling always has an impact on the pressure sub-problem and the properties of the respective matrices. This influence may be counter-productive for an efficient application of AMG.
There have been attempts to improve the robustness of AMG. Krylov-Deflation methods and Schwarz approaches, together with stronger AMG-smoothers, have been considered by Klie, et.al. and Clees and Ganzer, respectively [27, 60].

The pressure is approximately decoupled from the further unknowns, as the correct solution of these unknowns is not yet known in any iteration of CPR-AMG. Hence, the influence of the respective errors on the pressure approximation shall be reduced. As a full decoupling, for instance, via a Schur complement, is impossible for practical reasons, this decoupling is only approximated: the discrete pressure per grid cell is only decoupled from the further unknowns in this cell, as these couplings are expected to dominate. This could be achieved by cell- (or

block-)wise Schur complements, which leads to the *quasi-IMPES* method, as described, for instance, by Lacroix et.al. or Jiang [52, 63]. Both authors also suggest a comparable incorporation of physical information in order to make the local approximation of the full Schur complement meaningful in a physical sense. This approach is known as *true-IMPES*.
Different further decoupling approaches are described in the literature. Klie et.al. [59] considered the *Alternate Block Factorization* by Bank et.al. [9] and Lacroix, et.al. [63] introduced local QR decompositions for the decoupling. All the decoupling methods described so far have in common that the pressure problem is affected in a way that may drastically change the matrix properties that AMG requires.
Other decoupling methods have been proposed that do not potentially introduce new difficulties for AMG. Scheichl et.al. [90] suggested to compute an average pressure problem in order to decouple the pressure. Al-Shaalan, et.al. [5] described a method based on least-squares, where the pressure sub-problem is only scaled by constants.

However, with all these approaches there are situations observed where AMG fails. A recent example has been reported by Li et.al. [64] for thermal simulations.
Even if no new difficulties for AMG have been introduced by a matrix transformation, problems for multigrid methods may still occur. This results from the highly complex physics that are reflected by the linear systems and which cause the systems to not always fulfill the properties that multigrid seeks to exploit. In particular, the modeling of wells should be mentioned that can result in the pressure problems even being indefinite.

We should mention that also different methods for the pressure approximation in CPR are reported in the literature. For instance, Fung and Dogru [43] presented a way to drastically accelerate the convergence speed of incomplete factorizations for the pressure problem. By exploiting physical information, they used a matrix re-ordering that aims at appropriately capturing flow directions. This way, certain global information on the flow in the entire domain are incorporated in the factorization.
A recent further direction of research are *multiscale solvers* for the pressure problem, as, for instance, described by Wang, Hajibeygi and Tchelepi [104]. Here, a two level method is constructed in a way that is comparable to AMG. However, the transfer operators do not rely on matrix information only, but incorporate

information from a structured grid. In a sense, this can be seen as enriching an AMG approach with grid information.

Ultimately, the exploitation of a hierarchy is crucial to efficiently solve the linear systems that result from diffusion-dominated processes. As AMG methods construct their hierarchy independently of geometrical information, AMG-based solution methods are preferred by industrial simulation codes ([17, 19, 45, 50]). In the scope of this thesis, we will discuss how we can realize an efficient AMG-based solution approach that is robust and reliable. Furthermore, we will discuss how we can cover the outlined range of simulation models with such an approach.

We will exploit the fact that AMG can not only be used to solve linear problems for a single physical unknown, but that the AMG idea can also be extended to coupled systems with different physical unknowns. Such *System-AMG* approaches have been outlined by Ruge and Stüben [85] and are described in detail by Clees [26]. There have already been some attempts to employ such approaches for reservoir simulations: Clees and Ganzer [27] applied an AMG hierarchy for all unknowns from Black-Oil simulations and used alternating Schwarz techniques and strong AMG-smoothers to improve AMG's robustness.
Clearly, the matrices that describe coupled systems, or rather the sub-blocks within them that correspond to the different unknowns and the couplings in-between, need to satisfy certain properties for System-AMG to be efficient. Hence, due to the complex physics, the difficulties from CPR-AMG also apply here.

In this work, a particular focal point will be the robustness of the System-AMG approaches. On the one hand, this means that we need to select the components of System-AMG properly, according to the particular requirements of the respective simulation models. We will especially see that a simultaneous hierarchical treatment for different physical unknowns can be necessary.
On the other hand, we need to introduce some preparatory matrix transformations. These, in contrast to approximate decouplings from CPR-AMG approaches, will not introduce any artificial difficulties for AMG. On the contrary: we use these transformations to avoid the complex physics causing problems for System-AMG.

After presenting the robust applicability, we are going to investigate efficient solution approaches with the System-AMG framework to solve the full linear systems from reservoir simulations with multiple different physical unknowns. This framework is not only a convenient way to integrate the linear solution process into a

reservoir simulator, but can exploit all advantages of System-AMG.
We can first of all extend our solution approach to more sophisticated models with various physical unknowns. This way, we cover the full range of industrially relevant simulation models, starting from basic Black-Oil simulations, including compositional models and ending with sophisticated thermal and geomechanical applications.
Secondly, the full bandwidth of AMG-components is exploitable and we can optimize the System-AMG approach according to the typical matrix properties in reservoir simulations. Such optimizations can directly reflect the entire linear solution process and are not limited to only a sub-system. We will demonstrate this by optimizing the coarsening strategy of System-AMG.
Finally, System-AMG handles the linear systems that are directly derived from the simulation models and avoids any influences that are artificial and possibly problematic from AMG's perspective.

We will demonstrate the efficiency and robustness of the solution approaches with representative linear systems, mostly provided directly by industrial users. In fact, all problems that we have been concerned with can be solved in the described way, also various problems that could not be solved by AMG-based methods before.
That is, the resulting System-AMG approaches can be applied in a black-box fashion in a simulator, given that certain basic information regarding the type of model and the involved physical unknowns are made available, for all the types of models that we consider in this thesis.

The outline of this thesis is as follows:

As a first step, in **Chapter 2**, we will discuss the origin of the linear systems with a focus on the properties of the corresponding matrices. We will see how different aspects of the models are responsible for particular matrix properties, and under which conditions we can expect sub-matrices to be in matrix classes for which AMG works well. Our description starts with basic *Black-Oil* models, pointing out the influences of compressibility and wells. In these models, which still are heavily used in industrial reservoir simulations, we are only concerned with three types of fluid, namely oil, gas and water.
We continue with *compositional* simulations, where different chemical components are distinguished within the fluids. The properties of the linear systems under the *natural variable* and the *volume balance formulation* will be investigated and we

are going to discuss how they relate to the Black-Oil case.
Finally, we are going to discuss *thermal* and *geomechanical* models and analyze the matrix properties that are added, compared to the pure fluid flow simulations. Regarding geomechanics, we will outline the influence of mechanical forces on fractures and permeabilities. However, we limit our detailed discussion to the *poroelasticity* models, as we are only going to be concerned with such problems in subsequent sections.

In **Chapter 3**, we are going to describe a System-AMG approach for Black-Oil models under idealized assumptions on the properties of the linear systems. This starts with a brief introduction into AMG and System-AMG. We discuss which matrix properties are required in order to apply System-AMG in a robust way. From the previous chapter we know under which conditions certain properties can be expected for different sub-blocks of the full matrix. For the Black-Oil case we will see that the CPR method can be interpreted as a particular System-AMG approach. We will discuss the convergence properties of the solver approach under some model assumptions. We will see that the coupling between pressure and further unknowns is rather not crucial, in contrast to the reverse couplings and the properties of the pressure sub-problem itself.

Chapter 4 considers more realistic situations, where the ideal conditions from the previous chapter are not necessarily fulfilled. This is typically the case in practice. Instead of attempting to adjust the System-AMG approach to cope with all potential difficulties, which may not even be possible, we will describe a matrix transformation that can be used to ensure AMG's applicability. This transformation will exploit the entire information provided within the full system matrix and applies knowledge on the matrix properties that we have described in Chapter 2. We will see that System-AMG, in combination with this matrix transformation, can also handle Black-Oil systems that were originally challenging, or even impossible, to solve.

Based on the robust and efficient System-AMG approach for Black-Oil simulations, the **5th chapter** will extend this approach to compositional, thermal and geomechanical simulations. Here we exploit the strength of the System-AMG framework to be extendable to systems with additional physical unknowns. Especially in thermal and geomechanical simulations we will see that the ability of exploiting a hierarchical treatment, simultaneously for different unknowns, is beneficial.

After the System-AMG approach has been established also for sophisticated problems, in **Chapters 6**, we are going to discuss selected options to improve the computational performance of our solver. We can further accelerate the AMG approach, which we will demonstrate with an optimized coarsening process. We will find that we can exploit aggressive coarsening variants without loosing convergence speed.
We will also discuss performance improvements of the AMG smoother by exploiting special matrix properties in adaptive implicit simulations.

Practical aspects also include parallelization and implementational aspects of the linear solver. Our System-AMG approach utilizes incomplete factorizations, which are a significant performance bottleneck. These factorizations are inherently sequential processes and algorithmic compromises within their parallelization can cause severe convergence issues, especially for the challenging problems from reservoir simulations. For shared memory architectures, in **Chapter 7**, we will provide a parallelization approach that ensures equivalence to the sequential factorization.

Acknowledgments

I wish to express my deep gratitude to Dr. Klaus Stüben. On the one hand, I am grateful for his support regarding my PhD project and his efforts in proof-reading this thesis. On the other hand, I also appreciate his general support over the years since I started working in his group and would like to summarize it as: thank you for providing chances - or simply: Danke !

My sincere thanks also go to Prof. Dr. Ulrich Trottenberg for advising my PhD project and to Prof. Dr. Axel Klawonn for co-advising it. I appreciated their advises, especially during the writing period of this thesis. I am grateful to Prof. Hamdi Tchelepi, PhD, for his willingness of being the second co-advisor of this thesis.
I would also like to thank Prof. Dr. Ewald Speckenmeyer and Dr. Roman Wienands for participating in the committee of the defense of this thesis.

I am happy that I have had the chance of a PhD project in the SAMG group at the Fraunhofer Institute SCAI. This gave me the chance of learning a lot, also beyond the mere PhD research. I would like to thank the entire team, the current and the previous members, and my other colleagues at SCAI for the great

working atmosphere. My special thanks, apart from the department's head, Klaus Stüben, go to Dr. Hans-Joachim Plum and Arnold Krechel for various fruitful discussions and also for proof-reading parts of this thesis. I also appreciated Dr. Guy Lonsdale's support in proof-reading the introduction.

Being part of the SAMG team was especially great because it gave me the opportunity to meet many inspiring scientists all over the world, both in academia and industry. All the discussions that I have had with all these people have been a great experience. I am grateful to all of them.

I would like to name a few of these people, whom I am grateful for their particular support in the research that lead to this thesis.
I would like to thank John Ruge and Marian Brezina for several fruitful discussions around AMG. It was great to learn from such AMG experts.
My sincere thanks also go to the AD-GPRS team around Hamdi Tchelepi at the Stanford University. Several test cases in this thesis result from their simulator. I would like to especially thank Yifan Zhou, Denis Voskov, Timur Garipov, Sergey Klevtsov and Nicola Castelletto for their advise on the usage and coding with AD-GPRS.
I am happy that I was also allowed to present results with test cases from industrial users, namely, the Computer Modeling Group Ltd. in Calgary. I would especially like to thank Dave Collins and Geoffrey Brown for the close cooperation.

Last but not least, I am grateful to my family and friends. This work would not have been possible without the continuous love and support of my parents Jutta and Rainer. I would also like to express my sincere thanks for their support to all further members of my family: my sister Karoline, my aunts Martina and Gisela, my grand-father Kurt and the Schneider family.
Finally, I would like to thank my good friends Denis, Fred, Georg and Theresa for all the good times that we have spent together.

The General Purpose Research Simulator (GPRS) developed by the Reservoir Simulation Research Group (SUPRI-B) at Stanford University was used in this work.

Chapter 2

Linear Systems from Reservoir Simulation

The complex physics of reservoir simulation processes result in a need to consider various physical unknowns at the linear system level. Which unknowns, and how they are interconnected, depends on the concrete modeling approach. Knowledge of the concrete properties of the linear systems is necessary to design solver approaches in later chapters. Our description of reservoir simulation will therefore focus on these properties and how they depend on the physics. In our discussion we directly anticipate the objective of applying algebraic multigrid methods later on. Therefore, we particularly focus on relevant properties, like diagonal dominance, M-matrix properties, etc., in the different parts of the linear systems. Especially sub-problems that are described by M-matrices or essentially positive type matrices [15] are of particular interest for the design of an AMG-based solution method. We will discuss conditions under which these properties are fulfilled.

We are going to discuss details of modeling only to the extent they effect the linear systems. For further information we refer to the reservoir simulation book by Aziz and Settari [7] and further overview literature in [1, 6, 21].
We note that there is no unique modeling approach in reservoir simulation. While all approaches are based on the same fundamentals, they still differ in the concrete formulation to some extent. We consider all relevant model formulations. However, where different formulations result in comparable matrix properties, we will continue our discussion with the most convenient description.

2.1 Matrix Properties and Orderings

Before discussing reservoir simulation models and the resulting linear problems to be solved, we introduce some matrix properties that are of interest for the later discussion on linear solvers. Let arbitrary $A = (a_{ij}) \in \mathbb{R}^{m \times m}$ and let $\mathbb{1}$ the identity matrix, then we can state the following definitions:

Definition 2.1. Let arbitrary $v, w \in \mathbb{R}^m$, then $< v, w >_{std} = v^T w$ is the standard, or *Euclidean scalar product*. △

Definition 2.2. We call A *symmetric*, if $A = A^T$. △

Definition 2.3. $A^{sym} := \frac{1}{2}(A + A^T)$ is the *symmetric* part of A. △

Definition 2.4. We call A *positive definite*, if it is symmetric and has only positive eigenvalues. We also refer to it as an spd matrix (symmetric positive definite) and write $A > 0$. If A has only non-negative eigenvalues, we call it positive semi-definite and write $A \geq 0$. △

Definition 2.5. With an arbitrary matrix B of the same dimension as A, we have the *Löwner order* $A > B$ iff $A - B > 0$. We denote the positive semi-definite case with $\geq$. △

Definition 2.6. We call A *diagonally dominant*, if $\forall i \in 1, ..., m : |a_{ii}| > \sum_{j \neq i} |a_{ij}|$. If the condition holds only with $\geq$, A is called is weakly diagonally dominant. △

Definition 2.7. We call A *non-negative*, if $\forall i, j \in 1, ..., m : a_{ij} \geq 0$. △

Definition 2.8. A is a *Z-matrix*, if $\forall i, j \in 1, ..., m, i \neq j : a_{ii} > 0 \wedge a_{ij} \leq 0$. △

Definition 2.9. We call A an *M-matrix*, if it is a Z-matrix, its inverse A^{-1} exists and is non-negative. △

Definition 2.10. The *directed adjacency graph* $G_A = (V_A^G, E_A^G)$ is defined as follows: For each matrix row we have a vertex in V_A^G. We have an edge $e_{ij}^G \in E_A^G$ iff $a_{ij} \neq 0$.
We can construct a weighted graph by attaching the value a_{ij} as a weight to e_{ij}^G. △

Definition 2.11. We call A *irreducible*, if its directed adjacency graph is connected. △

Definition 2.12. We call a splitting $A = B - C$ with arbitrary matrices B, C *regular*, if B^{-1} exists and both B^{-1} as well as C are non-negative. △

Definition 2.13. $diag(A)$ is the *diagonal* of A. △

Definition 2.14. $\rho(A) = \{max|\lambda| : \lambda \text{ eigenvalue of } A\}$ is the *spectral radius* of A. △

Let us assume a discretization grid with n_{points} points that refer to the cell centers. $n_{unknowns}$ different physical unknowns are considered per point. Throughout this thesis we have to distinguish two types of matrix orderings for the linear systems $Ax = b$. We describe them by the order of the solution vector x. The *unknown-wise ordering* corresponds to:

$$x_{unknown-wise} = \begin{pmatrix} \text{unknown 1 at point 1} \\ \text{unknown 1 at point 2} \\ \vdots \\ \text{unknown 1 at point } n_{points} \\ \text{unknown 2 at point 1} \\ \vdots \end{pmatrix}. \tag{2.1}$$

The alternative *point-wise ordering* corresponds to:

$$x_{point-wise} = \begin{pmatrix} \text{unknown 1 at point 1} \\ \text{unknown 2 at point 1} \\ \vdots \\ \text{unknown } n_{unknowns} \text{ at point 1} \\ \text{unknown 1 at point 2} \\ \vdots \end{pmatrix}. \tag{2.2}$$

We will use the unknown-wise ordering mainly for theoretical discussions regarding the properties of the linear systems. However, when we discuss concrete linear solvers approaches in the later scope of this thesis, we will usually consider the point-wise ordering, as this is used in practice. Where the type of ordering is clear from the context, we will not explicitly mention this anymore. Both orderings describe the same linear problem and we can always assume an implicit reordering to be given.

2.2 Properties of Fluid and Stone

The fluid flow in porous media depends on both fluid properties and properties of the surrounding rock. At a later stage, we will deal with more sophisticated simulation models that additionally consider thermal or mechanical effects, which will require the consideration of further properties. Our discussion follows the considerations by Aziz and Settari [7], where also further details can be found.

We have to distinguish different types, or *phases*, of fluids. This distinction first of all is between hydrocarbons and water, because they cannot mix up. We moreover make a difference between liquid hydrocarbons (oil) and gaseous ones, because these can behave quite differently. That is, our model has to take three different phases α, β and γ into account. The fluid mixture is described by the *saturations* S_α, S_β and S_γ, which describe the relative amount of the respective phases in the entire fluid in a pore with $0 \leq S_\alpha, S_\beta, S_\gamma \leq 1$. In our following discussion, we assume all phases to be present in each pore of the reservoir to at least some extent, i.e., each saturation is non-zero. We will review the case of phases being absent at a later stage.
We assume the fluid to occupy the entire pore space in the rock. This volume consistency condition leads to the constraint:

$$S_\alpha + S_\beta + S_\gamma = 1. \tag{2.3}$$

In the following, with S we refer to the entire triple $\{S_\alpha, S_\beta, S_\gamma\}$.
When referring to properties of the entire fluid mixture, it is common to use the term *total* (e.g., total pressure, total density, etc.). The total analogue Y_{tot} of some property Y is defined as the average, weighted by the saturations:

$$Y_{tot} := S_\alpha Y_\alpha + S_\beta Y_\beta + S_\gamma Y_\gamma. \tag{2.4}$$

Usually, however, we consider the different properties per phase. We are going to describe the properties and balance equations exemplarily by means of phase α and note that the situation is analog with the other phases.

Because the fluid flow is driven by pressure differences, the *pressure* is an important physical unknown for our description. The different phases may have different pressures p_α, p_β and p_γ. This appears counter-intuitive at first glance, since all phases are located in the same pores. However, due to capillary forces and surface

tensions, the pressures differ for multiphase flows in porous media. But still the pressures of different phases within the same location are closely related. This is modeled by the saturation-dependent *capillary pressure* $p_{cap}^{\alpha\beta}$ that defines a linear equilibrium relation for the pressures of the phases α and β:

$$p_{cap}^{\alpha\beta} = p_\alpha - p_\beta. \tag{2.5}$$

Analog relations are used for α and γ, as well as β and γ. The capillary pressures depend on the distribution of the three phases within a pore. In the above equation, according to Aziz and Settari [7], $p_{cap}^{\alpha\beta}$ increases with increasing S_α and decreases with increasing S_β.

The fluid properties are described by the *viscosity* μ_α and the *density* ρ_α. The viscosity describes the inverse of the flowability under a given pressure and temperature, where the pressure's influence typically is small [7]. We will get back to the temperature's influence when considering thermal effects later. In contrast to the viscosity, the density is highly pressure dependent. It describes the mass per volume of the fluid, i.e. $\rho_\alpha := \frac{m_\alpha}{V_\alpha}$.

The concrete mixture of the fluid has an impact on the phases' flowability. For instance, the amount of water that is present in a pore of the rock influences how well oil can flow through this pore. It also makes a difference whether the surface of a pore is covered by water (water-wet) or not (oil-wet), as illustrated in Figure 2.1. These effects are modeled [1, 7] by the scalar *relative permeability* k_α^{rel}, a non-negative quantity that describes how well each phase can flow through a particular pore.

As it describes flowability, the relative permeability is comparable to what the inverted viscosity describes. Both properties often are combined to the term *mobility*:

$$\lambda_\alpha := \frac{k_\alpha^{rel}}{\mu_\alpha}. \tag{2.6}$$

In addition to the properties of the fluid, there is the porous medium the fluid is flowing through. A subsurface reservoir is not like a huge cavity where free flow is possible in all directions. It is more a sponge-like structure with interconnected pores, as sketched in Figure 2.1. To account for this structure, we have to introduce two additional quantities, the porosity and the permeability.

The *porosity* φ determines the amount of volume that is available for the fluid mixture, i.e., not consisting of stone. This implies $0 \leq \varphi \leq 1$.

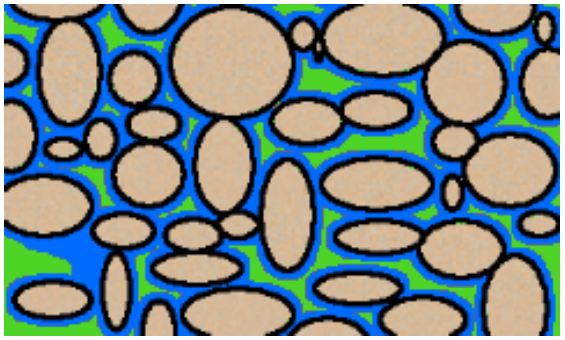


FIGURE 2.1: Two phases of fluid (green and blue) in a porous medium. The rock particles are colored like sand. The rock is "blue-wet", i.e., the green phase has no direct contact to the rock.

Not only the mere pore-volume is determining the fluid flow in reservoirs, but also the interconnection of the individual pores. It would be the most accurate way of modeling to fully resolve each pore and its connections. However, apart from the fact that the concrete pore structure of the sub-surface is typically not known in detail, this would result in giant models. The pore-volume is on a millimeter-scale, while petroleum reservoirs can extend kilometers in all space directions. Therefore, modeling the entire reservoir at the pore-scale is not feasible, even with modern computer architectures.

To nevertheless take the rock structure and the fact that flow is not possible in all directions equally well, into account, we introduce the *permeability* tensor $\vec{\vec{K}}$ [1, 7], also referred to as K. It determines the connectivity of the pore structure in the respective directions. The permeability is location dependent and may vary drastically over the reservoir domain, as exemplarily shown for the SPE10 problem in Figure 2.2. This makes the linear problems to be solved highly heterogeneous.

Moreover, the permeability at each point typically is anisotropic. If all entries per point were equal, the permeability field would be called isotropic. Because the permeability may still differ between different points, this still results in heterogeneous linear problems.

The permeability tensor is a simplification of the infinite number of possible flow directions in reality. However, since the pore structure of the sub-surface rock is only known approximately anyway, it is even not uncommon to allow for non-zero entries in K only along the diagonal. As soon as the discretization then is not perfectly aligned with the coordinate axes, i.e., the cells' normals point in different directions than the permeabilities, this leads to an incorrect discrete formulation of the fluid flux. Later, we will discuss multi-point flux approximations that can

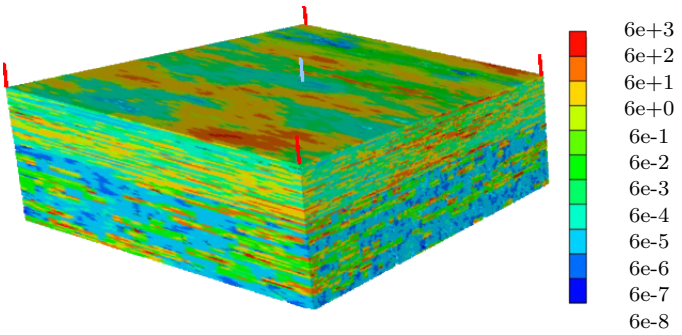


FIGURE 2.2: Highly heterogeneous permeability field of the SPE10 benchmark case. Shown is the permeability in millidarcy. The model uses one injector in the center (light blue) and four production wells (red) in its corners.
Picture from Computer Modeling Group Ltd., CMG.

overcome this problem, and their impact on the diffusion related parts of the linear systems.

What we have described so far are major properties that we will use throughout this thesis. The following remarks summarize alternative formulations that, however, lead to the same matrix structures:

Remark 2.15. *Instead of three, there might be only two phases being considered. This is, only hydrocarbons and non-hydrocarbons are distinguished and the amount of gas is taken into account by the hydrocarbon's density. Such a model is called* Dead-Oil model. *The linear systems are obtained as in the three-phase case, just the number of physical unknowns decreases.*
Also additional phases, like solids or coke from combustion, can be taken into account. We do not consider such additional phases in the scope of this thesis. △

Remark 2.16. *An alternative approach of taking the compressibility into account is to use a* formation volume factor B_α *instead of the density [21, 42].*
$B_\alpha := V_\alpha / V_{ref}$ *relates the volume* V_α *of the respective fluid with mass* m_α *in the reservoir with its volume* V_{ref} *at a reference state, e.g., surface conditions. There is a direct link between the formation volume factor and the density:*

$$\rho_\alpha = \frac{m_\alpha}{V_\alpha} = \frac{m_\alpha}{V_{ref} B_\alpha} = \frac{\rho_{ref}}{B_\alpha}.$$

From a linear solver's point of view, we have to deal with the same matrix properties, regardless whether a formation volume factor or the density is used. Hence, we do not consider modeling with a formation volume factor in this work. △

Remark 2.17. *Instead of using the saturations to describe the fluid mixture of the different phases, which gives a relative reference, we could also use a molar formulation. For each point N_α then describes the number of moles of phase α. We will get back to such approaches when discussing compositional models.* △

Remark 2.18. *Especially high pressures have an influence on the porosity. This can either be modeled by including geomechanical models in the simulation, or by a simplified pressure dependent porosity [1, 21]:*

$$\varphi = \varphi_{ref} e^{C_{rock}(p_{tot} - p_{ref})}. \tag{2.7}$$

$C_{rock} \geq 0$ denotes the rock's compressibility. We will consider the porosity to be constant for the moment and get back to poroelasticity models later on. △

2.3 Black-Oil Model

We start our discussion with reservoir models that only consider fluid flow and assume temperature and mechanical forces to be constant. We moreover assume oil to be one 'black' fluid, rather than decomposing it into its chemical components. A model that considers only the three phases oil, gas and water, is called Black-Oil (BO) model. It still is widely used in the oil industry. Moreover, more sophisticated modeling approaches, for instance, compositional and thermal ones, can be seen as an extension of the Black-Oil model.

Fluid flow is simulated based on the conservation of mass for the three phases. Following Aziz and Settari and Aziz, et.al. [6, 7], for phase α the mass balance is given by

$$0 = \underbrace{\frac{\partial}{\partial t}\varphi S_\alpha \rho_\alpha}_{\text{Accumulation Part}} + \underbrace{\nabla \rho_\alpha \vec{v}_\alpha}_{\text{Flux Part}} + \underbrace{\rho_\alpha q_\alpha}_{\text{Source Part}} . \tag{2.8}$$

It expresses that the mass of phase α that is accumulated in a point over time, is the result of in- and outflow of mass. In- and outflow can result from flux within the reservoir, which is described by the fluid velocity $\vec{v}_\alpha$, but it can also result from external sources ($q_\alpha < 0$, e.g., injection wells) and sinks ($q_\alpha > 0$, e.g., production wells).

The fluid velocity can be expressed in terms of the pressure gradient, as described

by Darcy's Law [7, 106] for fluids in porous media:

$$\vec{v}_\alpha = -\lambda_\alpha \vec{\vec{K}}(\nabla p_\alpha - g\nabla d), \tag{2.9}$$

where d is the depth and g the gravitational constant.
Combining the previous two equations leads to the mass conservation equation for phase α that is the basis for our further discussion:

$$0 = \underbrace{\frac{\partial}{\partial t}\varphi S_\alpha \rho_\alpha}_{\textit{Accumulation Part}} \underbrace{- \nabla \rho_\alpha \lambda_\alpha \vec{\vec{K}}(\nabla p_\alpha - g\nabla d)}_{\textit{Flux Part}} + \underbrace{\rho_\alpha q_\alpha}_{\textit{Source Part}} . \tag{2.10}$$

The objective of a simulator is to compute saturations and pressures for each grid cell, such that this mass balance equation is fulfilled for each phase at any time. We should emphasize that this equation is highly nonlinear, since various of the involved quantities depend on pressure and saturation.

The model would not be complete without appropriate boundary conditions. If the entire reservoir is considered for the model, then the permeability, via Darcy's Law, introduces a rather natural boundary condition. The permeability across the reservoir's boundary simply is zero and can be interpreted as a boundary condition for the pressure.
In addition, influences from sources and sinks, i.e. q_α, needs to be considered. We refer to our discussion in Section 2.4.
The saturations do not require boundary conditions. They only describe the amount of fluid of a particular phase per point of the reservoir domain. This by itself is independent of the distribution of the fluid at some neighboring point. However, we need some initial setting.

Remark 2.19. *Modeling the solution of compressed gas in the oil phase may result in a slightly different mass balance for the gas, as, for instance, described by Forsyth and Sammon [42]. This approach can reflect that, with increasing pressure, more and more gas becomes liquid and is dissolved in the oil phase. The amount of oily gas is described by the* oil-gas ratio R_{og}.

The gas that is dissolved in the oil phase behaves like the original oil phase. Therefore, the mass balance of gas has to involve dissolved and free gas, weighted by R_{og}:

$$\begin{aligned} 0 =& \frac{\partial}{\partial t}\varphi[R_{og}S_{oil}\rho_{oil} + S_{gas}\rho_{gas}] \\ & -\nabla(R_{og}\rho_{oil}\lambda_{oil}\vec{K}(\nabla p_{oil} - g\nabla d) + \rho_{gas}\lambda_{gas}\vec{K}(\nabla p_{gas} - g\nabla d)) \\ & +(R_{og}\rho_{oil}q_{oil} + \rho_{gas}q_{gas}). \end{aligned} \tag{2.11}$$

The basic structure of the gas phase's mass balance does not change. The same holds for the resulting linear problems to be solved, as we will see in Remark 2.24. For reasons of simplicity, we neglect the effect of dissolved gas in our further considerations. △

2.3.1 Partly Explicit Simulation Approaches

Before considering approaches that treat all unknowns implicitly, we review *partly explicit simulation approaches*. We do so, although we will not be concerned with such simulations in this thesis. These approaches have been developed first and some ideas carry over to fully implicit approaches. Therefore, it is worth reviewing them, namely the IMPES and volume-balance method. Both treat only the pressure unknown implicitly and differ regarding the further unknowns. By only treating the pressure implicitly, clearly, the time step size is limited for both methods.

In the *IMPES* (=IMplicit Pressure, Explicit Saturation) [13, 29] approach, a scalar differential equation for the pressure is considered. This is resulting from the mass balances, however, it treats the fluid as one single phase. This is, the total fluid properties are used and the total pressure is the unknown (cf. (2.4)). The distinction between the different phases is realized in a second step. The IMPES pressure equation reads as:

$$-\nabla\rho_{tot}\lambda_{tot}\vec{K}(\nabla p_{tot} - g\nabla d) = \rho_{tot}q_{tot}. \tag{2.12}$$

By using the previous time step's density, mobility and source term, the pressure PDE becomes a linear, elliptic differential equation. The source term induces a Dirichlet boundary condition at the well heads. Due to the elliptic nature of the PDE, algebraic multigrid methods are well-suited for solving the systems that

result from the discretization. The saturations are then updated explicitly via the mass balances (2.10), which result in strong limitations of the possible time step sizes. A way to improve this is provided by streamline approaches [10]. AMG already is successfully used in such simulators for solving the pressure problem.

In contrast to the IMPES-method, the *volume-balanced, or molar approach* aims at expressing the mass balances (2.10) in terms of moles N_α, rather than saturations S_α. This is particularly suited for simulation approaches where the individual oil components are considered, i.e., compositional simulations. We will get back to this aspect later.
As we do not have saturation unknowns in molar formulations, we cannot use (2.3) to ensure a volume-balance, but need to model it separately. This will give the pressure equation in such formulations. For details, we refer to Acs et.al. [4].

2.3.2 Fully Implicit Simulation Approaches

In contrast to the previous partly explicit simulation approaches, *fully implicit methods (FIM)* treat all physical unknowns implicitly. This allows for much larger time steps. Nevertheless, the concrete size of the time steps still has some limitation due to required accuracy and the required convergence speed of the linearization process.
Before linearizing, reservoir simulators discretize the nonlinear set of equations. Classical techniques for discretizing second and first order PDE operators can be used for this purpose. In the scope of this thesis, we consider finite difference approaches with all unknowns being defined at the cell centers. Other techniques, like finite volumes, result in linear systems with essentially the same properties. Details regarding such discretization techniques in reservoir simulations are, for instance, discussed by Moog [74].
Linearizing the discretized problem makes exploiting automatic differentiation schemes practicable. The application of such schemes is, for instance, described by Zhou [110]. Such schemes are used for the spacial derivatives. The time discretization of the accumulation term is realized with backwards differences at a time step size Δt.

The linearization typically is realized by Newton's method. This requires each of the unknowns to be associated with one of the equations. Based on only the discretized mass balances, we have an under-determined problem. For each of

the phases α, β and γ, we have one balance equation with two unknowns, the respective pressure and saturation. However, there are further equations involved. The capillary pressures (2.5) allow to express two pressures in terms of the third one. Only this single pressure needs to be solved for. In our discussion we choose the pressure of the first phase, p_α, and associate it with the first phase's mass balance equation. In this thesis, we assume α to be the oil phase, which we call the primary phase. We note that we could equally well choose any of the other pressures and associate it with any of the mass balance equations. As the mass balance equations involve the same terms, the matrix properties would, in principle, be the same.

We can use the saturation constraint (2.3) to express one saturation in terms of the other two. Since we have selected p_α to be the primary pressure, we choose $S_\alpha = 1 - S_\beta - S_\gamma$ and associate S_β and S_γ with the respective discretized mass balance equations.

Let us denote $[M_\alpha], [M_\beta]$ and $[M_\gamma]$ the discretized mass balance equations of the respective phase. With the discrete pressure and saturations, we have the vector of unknowns as $x = \begin{pmatrix} [p_\alpha] & [S_\beta] & [S_\gamma] \end{pmatrix}^T$. Starting from an initial guess x_0, the k-th iteration of Newton's method reads as:

$$x_k = x_{k-1} - \delta \text{ with } J(x_{k-1})\delta = \begin{pmatrix} [M_\alpha](x_{k-1}) \\ [M_\beta](x_{k-1}) \\ [M_\gamma](x_{k-1}) \end{pmatrix}, \tag{2.13}$$

where $J(x_{k-1})$ is the Jacobian matrix evaluated with the current solution vector[1]. We also refer to it simply as J. This matrix describes the linear system of equations that we have to solve. With our above choice of which unknown is associated with which equation, we have

$$J := \begin{pmatrix} \frac{\partial [M_\alpha]}{\partial [p_\alpha]} & \frac{\partial [M_\alpha]}{\partial [S_\beta]} & \frac{\partial [M_\alpha]}{\partial [S_\gamma]} \\ \frac{\partial [M_\beta]}{\partial [p_\alpha]} & \frac{\partial [M_\beta]}{\partial [S_\beta]} & \frac{\partial [M_\beta]}{\partial [S_\gamma]} \\ \frac{\partial [M_\gamma]}{\partial [p_\alpha]} & \frac{\partial [M_\gamma]}{\partial [S_\beta]} & \frac{\partial [M_\gamma]}{\partial [S_\gamma]} \end{pmatrix}. \tag{2.14}$$

[1]The Newton method sometimes is simplified and uses $J(x_0)$ in each iteration.

As we are dealing with two different types of unknowns, we distinguish the respective parts of the Jacobian:

$$\begin{aligned} A_{pp} &:= \begin{pmatrix} \frac{\partial[M_\alpha]}{\partial[p_\alpha]} \end{pmatrix} \quad A_{ps} := \begin{pmatrix} \frac{\partial[M_\alpha]}{\partial[S_\beta]} & \frac{\partial[M_\alpha]}{\partial[S_\gamma]} \end{pmatrix} \\ A_{sp} &:= \begin{pmatrix} \frac{\partial[M_\beta]}{\partial[p_\alpha]} \\ \frac{\partial[M_\gamma]}{\partial[p_\alpha]} \end{pmatrix} \quad A_{ss} := \begin{pmatrix} \frac{\partial[M_\beta]}{\partial[S_\beta]} & \frac{\partial[M_\beta]}{\partial[S_\gamma]} \\ \frac{\partial[M_\gamma]}{\partial[S_\beta]} & \frac{\partial[M_\gamma]}{\partial[S_\gamma]} \end{pmatrix}. \end{aligned} \tag{2.15}$$

A_{pp} refers to the derivative of the primary phase w.r.t. the pressure unknown. In the further discussion we also refer to it as *pressure sub-block* of the Jacobian. Analogously, A_{ss} results from the saturation derivatives of the secondary phases; the *saturation sub-block*. The structure and properties of A_{pp} and A_{sp}, as well as of A_{ss} and A_{ps} are closely related. The first set of matrices is resulting from pressure derivatives of the discrete mass balances, while the second one results from saturation derivatives. These derivatives essentially determine the properties of the matrices, which we will examine in more detail in the following. We will do so for an exemplary one dimensional case and see that the results carry over also to higher dimensional cases.

Remark 2.20. *Kwok [61] (Theorem 5.2) was able to show that there is a maximal time step size $\Delta t_{max} > 0$ such that the Jacobian (2.14) is non-singular under mild assumptions. Hence, there is a unique solution for the respective linear problems. In one dimension this even holds for any time step size.* △

2.3.2.1 Matrix Properties - 1D Case

For reasons of simplicity, we discuss the discretization of the mass balances and the computation of pressure and saturation derivatives in detail only for a one dimensional case. We use the discretization via finite differences, as described by Cardoso and Durlofsky [20]. Each cell i has two neighbors, $i+1$ and $i-1$. We will refer to the neighbors as $i \pm 1$. In each following equation, the notation $\pm$ indicates that either the plus $+$ is used at all occasions, or the minus $-$.

We assume our grid to be a chain of three dimensional cells of width h, as depicted in Figure 2.3. That is, the reservoir extents in only one dimension, but two cells i and $i \pm 1$ still share a face of size $A_{i,i\pm1} = A_{i\pm1,i}$. For our discussion, the faces shall have the same size A and the same shape for all pairs of cells. Different shapes and sizes can be treated accordingly.

All physical quantities are given at the cells' centers. However, to express the

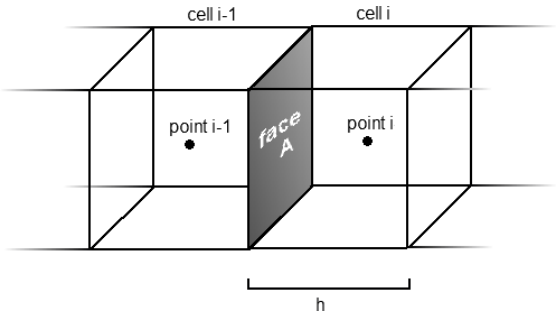


FIGURE 2.3: Exemplary single dimensional discretization grid.

flowability between two neighboring cells, *transmissibilities* need to be approximated at the cells' interfaces. We can describe the transmissibility for phase α between two cells as:

$$[T_\alpha]^n_{i\pm\frac{1}{2}} := \frac{K_{i,i\pm\frac{1}{2}}A}{h} \quad \frac{[\rho_\alpha]^n_{i\pm\frac{1}{2}}}{[\mu_\alpha]^n_{i\pm\frac{1}{2}}} \quad [k^{rel}_\alpha]^n_{i_\alpha^{upstream}}. \tag{2.16}$$

The block brackets $[\ ]^n_i$ shall indicate that we are referring to discretized quantities for point i and time step n.

For the saturation dependent relative permeability, according to Kwok [61], typically upstream weighting is used. That is, we decide which cell's quantity is used based on the pressure in the previous time step. As the pressures for different phases may differ, we do so phase-wise:

$$i_\alpha^{upstream} := \begin{cases} i \pm 1 & [p_\alpha]^{n-1}_{i\pm1} > [p_\alpha]^{n-1}_i \\ i & [p_\alpha]^{n-1}_i \geq [p_\alpha]^{n-1}_{i\pm1}. \end{cases} \tag{2.17}$$

For the pressure dependent density, either upstream weighting or averaging are common. That is, either $[\rho_\alpha]^n_{i\pm\frac{1}{2}} = [\rho_\alpha]^n_{i_\alpha^{upstream}}$ or $[\rho_\alpha]^n_{i\pm\frac{1}{2}} = \frac{1}{2}([\rho_\alpha]^n_i + [\rho_\alpha]^n_{i\pm1})$. This analogously holds for the viscosity. The permeability is always averaged, i.e. $K_{i,i\pm\frac{1}{2}} := \frac{1}{2}(K_{i,i\pm1} + K_{i\pm1,i})$, which results in a symmetrization of the permeability field.

With backwards differences for the time derivative in the accumulation term of the mass balance (2.10), the discretized equation for phase α in cell i at time n

reads as (other phases, cells and time steps follow analogously):

$$\begin{aligned}
[M_\alpha]_i^n := \quad & \tfrac{1}{\Delta t}([\varphi S_\alpha \rho_\alpha]_i^n - [\varphi S_\alpha \rho_\alpha]_i^{n-1}) && \Big\} =: [M_\alpha^{Accu}]_i^n \\
& -\tfrac{1}{hA}([T_\alpha]_{i-\frac{1}{2}}^n([p_\alpha]_{i-1}^n - [p_\alpha]_i^n - gd_{i-1} + gd_i) && \\
& \qquad + [T_\alpha]_{i+\frac{1}{2}}^n([p_\alpha]_{i+1}^n - [p_\alpha]_i^n - gd_{i+1} + gd_i)) && \Big\} =: [M_\alpha^{Flux}]_i^n \\
& +[\rho_\alpha q_\alpha]_i^n && \Big\} =: [M_\alpha^{Source}]_i^n.
\end{aligned} \tag{2.18}$$

The first row of the above expression, we denote it as $[M_\alpha^{Accu}]_i^n$, results from the accumulation part of the mass balance (2.10), while the last one, $[M_\alpha^{Source}]_i^n$, results from the source part. The two center rows, $[M_\alpha^{Flux}]_i^n$, result from the discretization of the fluid flux term, involving the transmissibilities at the cell's interfaces.
For reasons of simplicity, we assume a horizontal grid, i.e. $d_{i-1} = d_i = d_{i+1}$. Otherwise, the following argumentation would work analogously.

2.3.2.1.1 Pressure Related Blocks

We begin with the pressure sub-part A_{pp}, which is composed of the pressure derivatives of the accumulation, the flux and the source related summands of the discrete mass balance (2.18). So, there is $A_{pp} = A_{Accu,p} + A_{Flux,p} + A_{Source,p}$ with $A_{Accu,p} := \frac{\partial}{\partial[p_\alpha]}[M_\alpha^{Accu}]$, $A_{Flux,p} := \frac{\partial}{\partial[p_\alpha]}[M_\alpha^{Flux}]$ and $A_{Source,p} := \frac{\partial}{\partial[p_\alpha]}[M_\alpha^{Source}]$.
For the moment we just state that $A_{Source,p}$ is a diagonal matrix. We will see why this is the case when discussing external sources and sinks in Section 2.4.
To continue with $A_{Accu,p}$, we observe that, regarding cell i, only derivatives w.r.t. the pressure in cell i are relevant and $A_{Accu,p}$ is diagonal. The respective pressure derivative reads as:

$$\frac{\partial}{\partial[p_\alpha]_i^n}[M_\alpha^{Accu}]_i^n = \frac{1}{\Delta t}[\varphi S_\alpha]_i^n \frac{\partial}{\partial[p_\alpha]_i^n}\,[\rho_\alpha]_i^n. \tag{2.19}$$

Because the pressure derivative of the density is non-negative, $A_{Accu,p}$ is non-negative[2].

[2]If we took the rock's compressibility into account (cf. Remark 2.18), we would have the same result regarding $A_{Accu,p}$. The porosity (2.7) per cell only depends on the pressure in that cell and the pressure derivative is positive.

The situation is more involved for $A_{Flux,p}$, which results from the flux part of the discrete mass balance (2.18). Its derivative w.r.t. the pressure $[p_\alpha]_i^n$ reads as:

$$\begin{aligned}\frac{\partial}{\partial [p_\alpha]_i^n}[M_\alpha^{Flux}]_i^n =& \frac{1}{hA}([T_\alpha]_{i-\frac{1}{2}}^n + ([p_\alpha]_i^n - [p_\alpha]_{i-1}^n)\frac{\partial}{\partial [p_\alpha]_i^n}[T_\alpha]_{i-\frac{1}{2}}^n \\ &+ [T_\alpha]_{i+\frac{1}{2}}^n + ([p_\alpha]_i^n - [p_\alpha]_{i+1}^n)\frac{\partial}{\partial [p_\alpha]_i^n}[T_\alpha]_{i+\frac{1}{2}}^n).\end{aligned} \tag{2.20}$$

Analogously, for the derivatives w.r.t. the pressures $[p_\alpha]_{i\pm1}^n$ from the neighboring cells there is:

$$\frac{\partial}{\partial [p_\alpha]_{i\pm1}^n}[M_\alpha^{Flux}]_i^n = \frac{1}{hA}(-[T_\alpha]_{i\pm\frac{1}{2}}^n - ([p_\alpha]_{i\pm1}^n - [p_\alpha]_i^n)\frac{\partial}{\partial [p_\alpha]_{i\pm1}^n}[T_\alpha]_{i\pm\frac{1}{2}}^n). \tag{2.21}$$

In both cases, the transmissibility's derivative remains to be computed. It is obtained as:

$$\frac{\partial}{\partial [p_\alpha]_j^n}[T_\alpha]_{i\pm\frac{1}{2}}^n = \frac{K_{i,i\pm\frac{1}{2}} A [k_\alpha^{rel}]_{i_\alpha^{upstream}}^n}{h} \quad \frac{\partial}{\partial [p_\alpha]_j^n} \frac{[\rho_\alpha]_{i\pm\frac{1}{2}}^n}{[\mu_\alpha]_{i\pm\frac{1}{2}}^n}, \quad \text{with } j = i-1, i, i+1. \tag{2.22}$$

Because the pressure's influence on the transmissibility only results from density and viscosity, it makes a difference whether or not our model takes compressibility into account.

Incompressible case

For an incompressible fluid, the derivative of the transmissibility simply vanishes and we can state the following theorem:

Theorem 2.21. *In an incompressible model, assume the permeability between all neighboring cells to be positive. If there is at least one cell perforated by a well, i.e., a fluid sink or source, A_{pp} is an spd M-matrix. Without a well it was a zero row sum Z-matrix.*

Proof. In the incompressible case, $A_{Flux,p}$ is based only on the transmissibilities in (2.20) and (2.21), and not on their derivatives. This results in a weakly diagonally dominant zero row sum Z-matrix, i.e., $A_{Flux,p}$ is singular. We have $[T_\alpha]_{i\pm\frac{1}{2}}^n = [T_\alpha]_{i\pm1\mp\frac{1}{2}}^n$, which implies that $A_{Flux,p}$ is symmetric. The permeabilities' positivity ensures the irreducibility of $A_{Flux,p}$.

Because the derivative of the density vanishes, there is $A_{Accu,p} = 0$. $A_{Source,p}$ is a non-negative diagonal matrix in the incompressible case. The one well ensures at least one positive entry. For the moment, we accept this as given and refer to the

discussion in Section 2.4.1.2.
In summary, $A_{pp} = A_{Flux,p} + A_{Accu,p} + A_{Source,p}$ is a symmetric, weakly diagonally dominant, irreducible Z-matrix with at least one strongly diagonally dominant row. Hence, A_{pp} is an spd M-matrix ([83]). □

Corollary 2.22. *The same result analogously holds for the blocks of A_{sp}. With the capillary pressure (2.5) there, for instance, is $p_\beta = p_\alpha - p_{cap}^{\alpha\beta}$. Since the capillary pressure itself is not pressure dependent, the argumentation regarding A_{pp} directly carries over to the blocks of A_{sp}.* △

Remark 2.23. *If there was no pore-connection between cells i and $i+1$, i.e., a zero permeability, the entry $a_{i,i+1}$ in $A_{Flux,p}$ was zero. Hence, $A_{Flux,p}$ would not be irreducible anymore. However, it would simply consist of two blocks for which all above results hold. Hence, the matrix still would be a symmetric Z-matrix. Just the full matrix would not be irreducible, while the two blocks themselves would.*
If one of the independent sets of cells was not perforated by a well, the respective matrix block, and, hence, the entire matrix, was singular. Otherwise, however, all of the above carries over. △

Remark 2.24. *In the case of taking dissolved gas into account (cf. Remark 2.19) we also have to consider the pressure dependence of the oil-gas ratio R_{og}. Because more gas will be dissolved with an increasing fluid pressure, the pressure derivative of the oil-gas ratio is positive. So will be the pressure derivative of $R_{og}\rho_{oil}$. Thus, we have exactly the same matrix properties for those pressure related blocks of the Jacobian that are associated with the gas' mass conservation (2.11), as we had before.*
Therefore, in terms of matrix properties, it does not make a difference whether or not we take dissolved gas into account. △

Compressible Case

The pressure derivative of the transmissibility (2.22) does no longer vanish and we have to compute $\frac{\partial}{\partial [p_\alpha]_j^n} \left[\frac{\rho_\alpha}{\mu_\alpha}\right]_{i\pm\frac{1}{2}}^n$. We do not do so explicitly here, but leave it with a discussion on its sign. This is sufficient to understand the influence on the properties of $A_{Flux,p}$.
The pressure derivative of the density is positive. For the viscosity, according to Elsharkawy and Alikahn [39], we have to distinguish two cases. In the first case free gas is still present in cell i. An increasing pressure then results in gas being dissolved in the oil phase, which decreases the viscosity. Therefore, the pressure

derivative of the transmissibility is positive. In the second case there is no free gas present anymore that could be dissolved in the oil phase. The viscosity then increases with higher pressures. However, we mentioned earlier that the viscosity is temperature-dominated and the pressure only has a small impact. Thus, we can expect the density's pressure dependence to be strong enough so that the entire pressure derivative of the transmissibility still is positive.

Regarding the transmissibility's influence on $A_{Flux,p}$ it makes a difference how density and viscosity at the cell interfaces are discretized. We have to distinguish two cases:

1. Upstream weighting is used.
   That is, $[\rho_\alpha]^n_{i\pm\frac{1}{2}} = [\rho_\alpha]^n_{i_\alpha^{upstream}}$ with $i_\alpha^{upstream}$ from (2.17). In this case, $A_{Flux,p}$ is non-symmetric. For cell i the pressure derivative of the transmissibility at the interface to $i \pm 1$ either contributes to the diagonal of row i and to the off-diagonal for row $i \pm 1$, or vice versa.
   $A_{Flux,p}$ still is a Z-matrix: If $[p_\alpha]^n_{i\pm1} > [p_\alpha]^n_i$, then $\frac{\partial}{\partial[p_\alpha]^n_i}[T_\alpha]^n_{i\pm\frac{1}{2}} = 0$. That is, if the pressure difference $[p_\alpha]^n_i - [p_\alpha]^n_{i\pm1}$ is negative, the transmissibility's derivative vanishes. Hence, the diagonals of $A_{Flux,p}$ may only become 'more positive' and the off-diagonals 'more negative' than in the incompressible case. This moreover implies that $A_{Flux,p}$ still is irreducible. However, some rows may not be diagonally dominant now, while others are diagonally dominant and, in contrast to the incompressible case, $A_{Flux,p}$ is not necessarily singular.

2. Averaging is used.
   That is, $[\rho_\alpha]^n_{i\pm\frac{1}{2}} = \frac{1}{2}([\rho_\alpha]^n_i + [\rho_\alpha]^n_{i\pm1})$. The derivative of the transmissibility is multiplied with the pressure difference of cell i and its neighbor $i \pm 1$. Compared to the incompressible case, the coupling between i and $i \pm 1$ becomes smaller, while the one from $i \pm 1$ to i becomes bigger, or vice versa. This implies that off-diagonals could become positive and the Z-matrix property is not guaranteed anymore. Finally, $A_{Flux,p}$ is singular, since all entries per row sum up to zero, and it may not be weakly diagonally dominant in all rows.

We do not further discuss the case of averaging, since upstream weighting is commonly used in practice [61]. The fact that $A_{Flux,p}$ still is a Z-matrix then, enables us to state the following theorem on A_{pp}:

Theorem 2.25. *In a compressible model, assume no fluid sources (e.g., injection wells) to be present and the permeability between all neighboring cells to be positive. Assume upstream weighting to be used for all quantities defining the transmissibility (2.16). Then* $\exists \Delta t_{max} > 0 : \forall \Delta t \leq \Delta t_{max}$ A_{pp} *is an M-matrix. We discuss the impact of injection wells in Section 2.4.*

Proof. As discussed above, under the given assumptions, $A_{Flux,p}$ is an irreducible Z-matrix. $A_{Accu,p}$ is a diagonal matrix with positive entries from (2.19). Δt can be chosen such that $A_{Flux,p} + A_{Accu,p}$ is weakly diagonally dominant with at least one diagonally dominant row. With only production wells, $A_{Source,p}$ is diagonal with non-negative entries, as we will see in Section 2.4.1.2.
The further argumentation is as in Theorem 2.21. □

Corollary 2.26. *As with Corollary 2.22 in the incompressible case, the above argumentation carries over to the blocks of* A_{sp} *also in the compressible case.* △

Remark 2.27. *The situation again is the same if we took dissolved gas into account (cf. Remark 2.24).* △

2.3.2.1.2 Saturation Related Blocks

For the saturation part A_{ss} we only consider the upper-left block A_{s1s1}, resulting from $\frac{\partial}{\partial[S_\beta]}[M_\beta]$. The other three blocks are computed analogously. We decompose this block into $A_{s1s1} = A_{Accu,s1s1} + A_{Flux,s1s1}$ with the individual summands resulting from the saturation derivatives of the respective parts of the discrete mass balance (2.18): $A_{Accu,s1s1} := \frac{\partial}{\partial[S_\beta]}[M_\beta^{Accu}]_i^n$ and $A_{Flux,s1s1} := \frac{\partial}{\partial[S_\beta]}[M_\beta^{Flux}]_i^n$. In our later discussion we will see that the source terms from wells are independent of the saturations. Therefore, no $A_{Source,s1s1}$ needs to be considered here. We also do not need to distinguish the compressible and incompressible case. Neither density, nor viscosity are saturation dependent. This also holds for the gas-oil ratio. Hence, it does not make a difference whether the simulation takes dissolved gas into account.
For the flux part, $A_{Flux,s1s1}$, there are two influences. The saturation derivative of the transmissibility and the derivative of the capillary pressures from expressing phase β's pressure in terms of the primary pressure: $[p_\beta]_i^n = [p_\alpha]_i^n - [p_{cap}^{\alpha\beta}]_i^n$. For

the derivative of the flux part w.r.t. the saturation $[S_\beta]_i^n$ we obtain:

$$\begin{aligned}\frac{\partial}{\partial[S_\beta]_i^n}[M_\beta^{Flux}]_i^n =& \frac{1}{hA}(([p_\alpha]_i^n - [p_\alpha]_{i-1}^n - [p_{cap}^{\alpha\beta}]_i^n + [p_{cap}^{\alpha\beta}]_{i-1}^n)\frac{\partial}{\partial[S_\beta]_i^n}[T_\beta]_{i-\frac{1}{2}}^n \\ &+ ([p_\alpha]_i^n - [p_\alpha]_{i+1}^n - [p_{cap}^{\alpha\beta}]_i^n + [p_{cap}^{\alpha\beta}]_{i+1}^n)\frac{\partial}{\partial[S_\beta]_i^n}[T_\beta]_{i+\frac{1}{2}}^n) \\ &+ \frac{1}{hA}([T_\beta]_{i-\frac{1}{2}}^n + [T_\beta]_{i+\frac{1}{2}}^n)(-\frac{\partial}{\partial[S_\beta]_i^n}[p_{cap}^{\alpha\beta}]_i^n).\end{aligned} \tag{2.23}$$

For the saturations $[S_\beta]_{i\pm1}^n$ of the neighboring cells there is:

$$\begin{aligned}\frac{\partial}{\partial[S_\beta]_{i\pm1}^n}[M_\beta^{Flux}]_i^n =& \frac{1}{hA}(-([p_\alpha]_{i\pm1}^n - [p_\alpha]_i^n - [p_{cap}^{\alpha\beta}]_{i\pm1}^n + [p_{cap}^{\alpha\beta}]_i^n)\frac{\partial}{\partial[S_\beta]_{i\pm1}^n}[T_\beta]_{i\pm\frac{1}{2}}^n) \\ &+ \frac{1}{hA}[T_\beta]_{i\pm\frac{1}{2}}^n\frac{\partial}{\partial[S_\beta]_{i\pm1}^n}[p_{cap}^{\alpha\beta}]_{i\pm1}^n.\end{aligned} \tag{2.24}$$

According to Aziz and Settari [7], we have $\frac{\partial}{\partial S_\beta}p_{cap}^{\alpha\beta} \leq 0$. Consequently, the contributions of the respective terms from (2.23) and (2.24) on $A_{Flux,s1s1}$ are positive on the diagonal and negative on the off-diagonals.
For the transmissibilities (2.16) we have to compute the saturation derivative of the relative permeability:

$$\frac{\partial}{\partial[S_\beta]_j^n}[T_\beta]_{i\pm\frac{1}{2}}^n = \frac{K_{i,i\pm\frac{1}{2}}A[\rho_\beta]_{i_\alpha^{upstream}}^n}{h[\mu_\beta]_{i_\beta^{upstream}}^n}\ \frac{\partial}{\partial[S_\beta]_j^n}\ [k_\beta^{rel}]_{i_\alpha^{upstream}}^n \quad \text{with} \quad j = i-1, i, i+1. \tag{2.25}$$

A phase has a higher relative permeability, the higher its saturation is [7]. Hence, for the derivative of phase β's transmissibility w.r.t. S_β there is $\frac{\partial}{\partial[S_\beta]_j^n}[k_\beta^{rel}]_{i_\beta^{upstream}}^n \geq 0$. The phase-wise upstream discretization of the relative permeability ensures the contribution to $A_{Flux,s1s1}$ to be positive in the diagonals and negative in off-diagonals. This is, if $([p_\beta]_{i-1}^n - [p_\beta]_i^n) = ([p_\alpha]_{i-1}^n - [p_\alpha]_i^n - [p_{cap}^{\alpha\beta}]_{i-1}^n + [p_{cap}^{\alpha\beta}]_i^n) < 0$, then $\frac{\partial}{\partial[S_\beta]_{i-1}^n}[T_\beta]_{i-\frac{1}{2}}^n = 0$.
Together with the above influence from capillary pressures this yields $A_{Flux,s1s1}$ to be a Z matrix. However, it does neither have to be diagonally dominant nor irreducible.

Secondly, there is the accumulation part. Per cell, this only depends on quantities in that cell and $A_{Accu,s}$ is a diagonal matrix. We obtain:

$$\frac{\partial}{\partial [S_\beta]_i^n} [M_\beta^{Accu}]_i^n = \frac{1}{\Delta t} [\varphi \rho_\beta]_i^n. \tag{2.26}$$

Theorem 2.28. $\exists \Delta t_{max} > 0 : \forall \Delta t \leq \Delta t_{max} : A_{s1s1}$ *is an M-matrix.*

Proof. Δt_{max} can be chosen so that $A_{Accu,s}$ ensures A_{s1s1} to be diagonally dominant in each row. Then ([83]), it is not only a Z-matrix, but an M-matrix. □

Remark 2.29. *In all cases we know of, due to the influence of* $\frac{1}{\Delta t}$, A_{s1s1} *is strongly diagonally dominant.* △

Remark 2.30. *The properties of the off-diagonal blocks of* A_{ss} *depend on how the saturation of phase* γ *influences the relative permeability of phase* β*, and vice versa. If we assume any relative permeability only to depend on those saturations where the respective derivative is positive, then also the off-diagonal blocks of* A_{ss} *are Z-matrices. Following Aziz and Settari ([7], Section 2.7.2.2) it is even reasonable to expect the relative permeability of gas and water to depend only on the respective saturations.* k_{oil}^{rel} *can be expressed as a combination of both.*
The diagonal dominance of the off-diagonal blocks of A_{ss} *depends on how the accumulation parts of one phase depend on the saturations of another one. In any case, regarding cell* i*, the diagonal block* $[A_{ss}]_{ii}$ *in the point-wise numbering, (drastically) dominates the off-diagonal blocks* $[A_{ss}]_{ij}$*.* △

Remark 2.31. *The blocks of* A_{ps} *are computed analogously to the diagonal blocks of* A_{ss}*. Because we are directly considering the mass balance of phase* α *here, we do not have to deal with capillary pressures. However, this in principal does not have an impact on the matrix properties.*
There are just two differences. Because S_α *is expressed as* $1 - S_\beta - S_\gamma$*, a multiplication with* -1 *is involved. Hence, the blocks of* $-A_{ps}$ *are M-matrices, not those of* A_{ps}*. Secondly, as both blocks involve terms resulting from the accumulation, both typically are strongly diagonally dominant.* △

2.3.2.2 Matrix Properties - 2D and 3D Case

In the two and three dimensional case we have to be aware of the fact that the permeability tensor might not be aligned with the grid directions. This especially

holds in the case of unstructured grids, where the flux approximation by using a diagonal permeability tensor cannot be accurate anymore. However, in the interest of simplicity of the model, this often still is the method of choice, especially if the miss-alignment was not too strong. Many of today's reservoir simulators proceed this way and we will only be concerned with such problems in the scope of this thesis.
However, more sophisticated flux approximations can be applied in order to gain a more precise prediction of the fluid flow. We give a brief overview on possible influences on the matrix properties.

2.3.2.2.1 Two Point Flux Approximation TPFA

In the TPFA approach a possible miss-alignment of the permeability tensor with the grid is ignored. The flux between two adjacent grid cells i and j is assumed to be influenced only by these two cells and the Jacobian matrix is obtained just as in the one dimensional case. The number of matrix entries per row increases with the number of entries of K. However, the entries themselves are obtained just as in the one-dimensional case and we have exactly the same matrix properties.

2.3.2.2.2 Multi Point Flux Approximations MPFA

Multi point flux approximations (MPFA) [2, 3] have been developed that are able

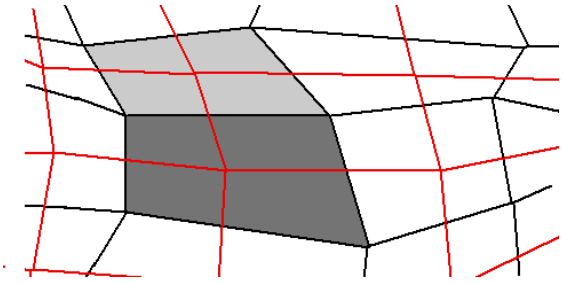


FIGURE 2.4: Simplified visualization of MPFA. The discretization grid is given in black lines, with cells i and j being dark and light gray, respectively. The dual grid is plotted in red lines. All cells of the discretization grid within a dual cell might be considered for the flux between cells i and j across the sub-interface within that dual cell.

to capture the flux more accurately. Later on, we will not be concerned with linear problems resulting from such discretizations. However, we briefly review these approaches and possible impacts on the pressure related matrix blocks.

The idea is to describe the flux between cells i and j not only by properties of these two cells, but to also involve surrounding cells. This way, the accuracy of the discretization becomes independent of whether or not the permeability is aligned with the discretization grid. An accurate discretization with this regard is called K-orthogonal.
The decision on which cells are considered to predict the fluid flow between cells i and j is made in a dual grid. It determines which further cells might be involved in the description of the fluid flow across an interface. Figure 2.4 visualizes such a situation. There are various approaches reported in the literature on how to particularly adjust the discretization, and this is still a field of active research. We refer to Aavatsmark's description [2] and the references therein for a detailed discussion.

In order to gain an idea on possible impacts on the linear systems, we will use an abstract expression of MPFA. This is described, for instance, by Zhou [110] and was already introduced by Aavatsmark [2]. We only consider the case of upstream weighting being used for the discretization of density and mobility. Then the flux of phase α from cell i to j in time step n is described by

$$f^{ij\alpha}_{MPFA} := \sum_{m=1}^{n^{ij}_{MPFA}} [\rho_\alpha \lambda_\alpha]^n_{i_\alpha^{upstream_{ij}}} [T^{MPFA}_{ij}]^n_{i,j,k^{ij}_m} ([p_\alpha]^n_{k^{ij}_m} - g d_{k^{ij}_m}). \tag{2.27}$$

With the respective formulations for all cells that are adjacent to cell i, we replace the flux part of the discrete mass balance, i.e. $[M^{Flux}_\alpha]^n_i$. We limit our discussion here to the flux from i to j, for exemplary i and j. $n^{ij}_{MPFA} \geq 2$ denotes the number of cells that are involved in obtaining the transmissibility for this flux. The indices of these cells are given by the vector k^{ij}. From the construction of the flux discretization we have:

- In order to ensure mass balance, there is $\sum_{m=1}^{n^{ij}_{MPFA}} [T^{MPFA}_{ij}]^n_{i,j,k^{ij}_m} = 0$
- Since both cells at the respective interface need to be incorporated, without loss of generality we assume $k_1 = i$ and $k_2 = j$
- As we are considering the flux from i to j, with $k_1 = i$ and $k_2 = j$, we have $[T^{MPFA}_\alpha]^n_{i,j,k_1} \geq 0$ and $[T^{MPFA}_\alpha]^n_{i,j,k_2} \leq 0$
- T^{MPFA}_{ij} is not pressure dependent. It is based on $\vec{\vec{K}}$.

In the special case of $n^{ij}_{MPFA} = 2$, the first and third of the above statements imply that the TPFA method is expressed in terms of this MPFA formulation and

the matrix properties would not change compared to what we have discussed so far. However, for the case $n^{ij}_{MPFA} > 2$ there generally will be an impact on the matrices, as the flux part of the discrete mass balance changes. That is, we have to expect changes in the properties of $A_{Flux,p}$ and $A_{Flux,S}$.
From a solver's perspective, due to its diagonal dominance, the modified flux approximation is less critical regarding A_{ss}. Therefore, we will only discuss the situation for A_{pp}, or $A_{Flux,p}$. We need to consider the derivatives of (2.27) w.r.t. the discretized pressures in cells i, j and any l, where cell l contributes to the transmissibility between i and j. We anticipate that the derivatives w.r.t. the pressures in cells i and j will result in the same matrix properties as in the TPFA case. There will be a positive influence on the diagonal and a negative one on the off-diagonal. This at least holds if we adjust the upstream weighting from (2.17) to the new situation of more than two cells contributing (see, for instance, [2]).
The situation is different for the derivative w.r.t. the pressure in cell l:

$$\frac{\partial f^{ij\alpha}_{MPFA}}{\partial [p_\alpha]^n_l} = [\rho_\alpha \lambda_\alpha]^n_{i^{upstream_{ij}}_\alpha} [T^{MPFA}_{ij}]^n_{i,j,l}. \tag{2.28}$$

This gives an additional non-zero entry in the i-th matrix row, i.e., a_{il} in $A_{Flux,p}$. This additional entry, depending on $[T^{MPFA}_{ij}]^n_{i,j,l}$, can either have a positive or a negative sign. In the latter case, this additional entry does not change the Z-matrix property of $A_{Flux,p}$ and all previous results carry over. However, in general there will be positive entries. We then loose the Z-matrix, and M-matrix, property of A_{pp}.

This does not necessarily need to be a problem for the application of AMG methods. There are also other classes of matrices that are suited for the application of AMG. One of them are essentially positive type matrices.

Definition 2.32. (From Brandt [15])
An arbitrary spd $m \times m$ matrix $A = (a_{ij})$ is called to be of *essentially positive type*, if there exists some $c > 0$ such that for all $e \in \mathbb{R}^m$ there is:

$$\sum_{i,j} (-a_{ij})(e_i - e_j)^2 \geq c \sum_{i,j} (-a^-_{ij})(e_i - ej)^2, \tag{2.29}$$

where $a^-_{ij} = \begin{cases} a_{ij} & \text{if } a_{ij} < 0 \\ 0 & \text{if } a_{ij} \geq 0 \end{cases}$. △

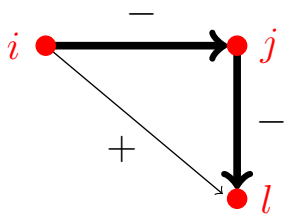


FIGURE 2.5: Excerpt of the weighted adjacency graph of an essentially positive type matrix. For the positively weighted vertex from i to l, there is an 'alternative' path i, j, l with much stronger, negatively weighted vertices.

Each positive off-diagonal in an essentially positive type matrix is dominated by negative ones. That is, if a_{il} was positive, in the weighted adjacency graph we could find a path from i to l via edges with strong negative weights. These negative entries are much stronger than the positive a_{il}. Figure 2.5 illustrates this situation. AMG can be shown to work properly for such matrices. See, for instance, Stüben [95].

For typical MPFA discretization schemes, we can expect the pressure sub-matrix A_{pp} (and analogously the blocks in A_{sp}) to be of essentially positive type. The positive off-diagonal entries only result from corrections of the transmissibility to ensure K-orthogonality of the discretization. We can still expect the influences of cells i and j to dominate the flux between these two cells. This implies that we can expect a negative off-diagonal entry per matrix row that should be (much) stronger than any positive one: As $A_{Flux,p}$ in the TPFA case was irreducible, in the MPFA case, for any positive a_{il}, we can find a path from i to l in the adjacency graph along vertices that correspond to strong negative values.
This coincides with practical experience: Let us exemplarily consider Case07 from ADGPRS' testsuite, a compositional Black-Oil model with MPFA discretization on a grid of 17550 cells. About a third of the off-diagonal entries in A_{pp} is positive (in total, as well as per row). However, the negative entries per row dominate the positive ones by a factor of 20.
In the scope of this thesis we are only concerned with linear problems resulting from a TPFA discretization. We are therefore not further investigating the situation in the MPFA context. We just note that AMG could be applied equally well where some matrix does not fulfill the M-matrix properties, but is of essentially positive type.

2.4 External Sources and Sinks

We have not yet considered the modeling of external sources and sinks and simply accepted some results as given. The external sources have a twofold impact on the properties of the pressure related matrix parts A_{pp} and A_{sp} (cf. Figure 2.6):

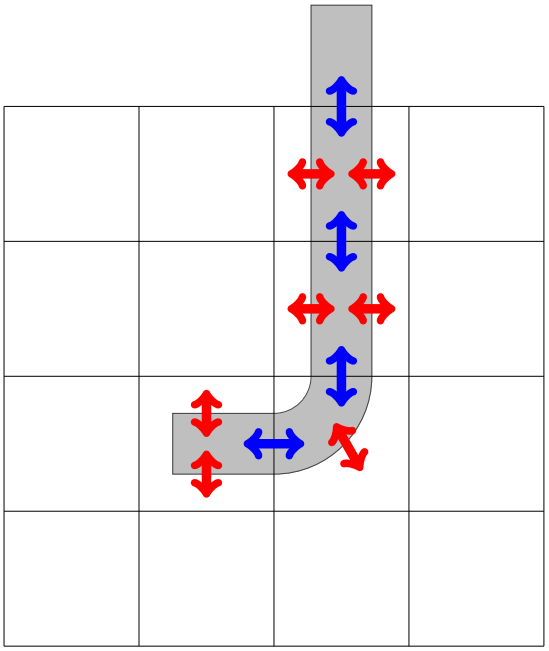


FIGURE 2.6: Schematic visualization of a well-bore's influence in a 2D structured grid: Interaction between well and reservoir (red arrows), as well as interactions within the well (blue arrows) have to be taken into account.

On the one hand, there is the influence from a well on the reservoir. This results in source terms influencing the respective matrices. When discussing AMG approaches, we will see that these influences can be quite severe.
On the other hand, the fluid flow within the well-bore needs to be modeled. This results in additional equations that describe the pressure within the well.
Finally, we note that external sources and sinks do not only need to be wells. Also fractures and chemical reactions can be modeled in such a way. The impacts on the linear systems, however, in principle are comparable to the ones from wells.

2.4.1 Source Term Resulting from Wells

In the discretized model, a well perforates several cells of the grid. All these cells have to reflect the presence of the well by the source term q_α that results from a

well model. As the source term is pressure-dependent, the respective rows of A_{pp} and A_{sp} are affected.

2.4.1.1 Well Model

The source term describes the flux of phase α between well and cell, resulting from the pressure difference. There are various approaches described in the literature to also model horizontal wells, complex well layouts, in-well facilities, etc. However, all these models are based on Peaceman's well model [81] for vertical wells. From a linear solver's point of view, all these models result in comparable impacts on the linear systems to be solved. A detailed description of the well model and some extensions is given by Chen and Zhang [23].

To gain an idea of the model, we consider a well in a two-dimensional reservoir, as

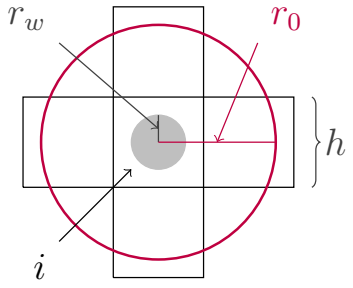


FIGURE 2.7: Visualization of Peaceman's well model in a horizontal grid. Cell i is the cell in the center, which is perforated by the vertical well of radius r_w (light gray). The well pressure is assumed to effect any point within the red circle of radius r_0. From this setting, the source term for cell i can be obtained.

in Figure 2.7. We use a structured grid with a uniform cell size h. We assume a well perforates a cell centrally and the well-bore of radius r_w is completely surrounded by the cell. The well is expected to have an influence on its surrounding in a radial way, decreasing with the distance from the well-bore. The description of this influence was found experimentally. There is a distance r_0, beyond which there is no further influence from the well. That is, the fluid flux beyond r_0 is resulting from the phase velocity only, which is described by Darcy's Law (2.9). These assumptions are the basis for turning the radial influence from the well into a description of the source term with a so-called *well-index*. We refer to Peaceman or Chen and Zhang [23, 81] for details.

The discrete source term from Peaceman's and related (e.g., horizontal wells, etc.) models reads as:

$$[q_\alpha]_i^n = WI_i[\lambda_w]_i^n([p_\alpha]_i^n - [p_w]_i^n). \tag{2.30}$$

It relates the cell pressure $[p_\alpha]_i^n$ to the well pressure $[p_w]_i^n$. As the well-bore is not a porous medium, but free flow is possible in all directions, the phases' pressures are not distinguished in the well. The transmissibility between well and cell is given by the well-index WI_i, which has a role comparable to K. It is obtained based on the well's geometry and the geological settings, independently of pressure and saturation. The well-mobility $[\lambda_w]_i^n$ is the analogue of the mobility $[\lambda_\alpha]_i^n$. It depends on the pressure in the well.
Both well-index and well-mobility are positive. The sign of the source term is determined by the pressure difference between the reservoir and the well. If this is positive, i.e., the well-pressure is lower than in the reservoir, the well produces. With a negative pressure difference, the well injects.
We note that indirect influence on cells next to a perforated one is reflected by the Darcy flux. That is, we need to consider a source term only for those cells that are perforated by a well.

2.4.1.2 Impact on the Matrix Properties

With the source term (2.30) we can compute $A_{source,p}$. From our above considerations it is clear that for cell i we only have to consider the derivative w.r.t. the pressure $[p_\alpha]_i^n$ and $A_{source,p}$ is a diagonal matrix. Hence, it does not make a difference whether we consider a one, two or three dimensional model. For the pressure derivative in cell i, here for phase α, we have

$$\frac{\partial}{\partial [p_\alpha]_i^n}[M_\alpha^{Source}]_i^n = [\rho_\alpha]_i^n WI_i[\lambda_{well}]_i^n + [q_\alpha]_i^n \frac{\partial}{\partial [p_\alpha]_i^n}[\rho_\alpha]_i^n. \tag{2.31}$$

The first summand of the above expression is non-negative, because density, mobility and the well-index are non-negative. Hence, this summand results in some non-negative values on the diagonal of $A_{Source,p}$. Regarding the properties of A_{pp} and A_{sp}, and the conditions for them to be M-matrices (cf. Theorems 2.21 and 2.25), nothing changes. In an incompressible model, the second summand of (2.31) simply vanishes.
The situation is different with a compressible model. Then the derivative of the density does not vanish, and the second summand becomes relevant. Since the

derivative of density w.r.t. pressure is positive, the sign of the source term decides on whether the second summand is positive or negative. Consequently, a production well, i.e. $q_\alpha > 0$, also in a compressible model results in positive diagonal entries in $A_{Source,p}$.
In contrast to this, injection wells contribute with negative values to the diagonal of $A_{Source,p}$. Consequently, in $A_{pp} = A_{Accu,p} + A_{Flux,p} + A_{Source,p}$ there may be several rows that violate diagonal dominance - even drastically. In the worst case, there might even be negative diagonal entries. We will see that this can have a drastic impact on the applicability of AMG methods.

2.4.2 Well Equations

So far, we have investigated the well's impact on the linear systems that result from the description of multi-phase flow within the reservoir. We have not yet considered the flow in the well-bore itself. Describing this flow is necessary to express the well's pressure $[p_w]_i^n$ in a particular cell, which is needed to compute the source term (2.30). There are two models that are commonly used, the *standard* and the *multi-segment well model*. Moreover, it makes a difference how a well is controlled at the surface[3]. This could either be *bottom-hole-pressure (BHP)*, where the pressure at the well's head is kept at a given value, or it could be *rate controlled*, where the flux rate at the well's head is under control.
The type and the number of the well equations are determined by which model and which type of control is used. We give a brief review in the following and refer to Jiang [52] for a detailed discussion.

2.4.2.1 Standard Well Model

In the standard well model, the well is discretized according to the reservoir grid. That is, the part of the well that perforates a reservoir cell, forms a well cell. Hence, the model directly describes $[p_w]_i^n$. It assumes the free flow within the well-bore to make the fluid decomposition into different phases negligible. Pressure differences in well cells then only result from gravitational effects. If we additionally neglect the compressibility within the well-bore, the well pressure in cell i can directly be

[3]Which flow rates and / or pressures are physically reasonable is determined by production engineering models. We refer to Econimides et.al. [37] for details.

obtained from a reference pressure:

$$[p_w]_i^n = \sum_{k=\text{cells from } i \text{ to ref.}} (d_k - d_{k-1}) + [p_w]_{ref}^n. \tag{2.32}$$

The reference pressure $[p_w]_{ref}^n$ depends on the type of control. With a BHP controlled well it can be prescribed and the well equations can be seen as a Dirichlet boundary condition that can be eliminated from the system rather easily. Otherwise, the reference pressure has to be computed from the reference flow rate and the linear system contains an additional well equation.

2.4.2.2 Multi Segment Well Model

Complex wells may be characterized by a varying diameter within the well-bore, and may also involve in-well facilities, e.g. velvets. Pressure differences in the well-bore now do not only result from gravitation. In the multi segment well model, the well therefore is modeled by different segments, a subdivision of the well-bore that is independent of the reservoir grid. This is described in detail by Jiang [52] and mentioned by Coats [28]. It leads to a generalized equation for the pressure difference between two well segments i and j:

$$[p_w]_i^n - [p_w]_j^n = \Delta_{ij}^p. \tag{2.33}$$

We can use Δ_{ij}^p to model gravitational, capillary and further effects on the pressure. In addition to the well's pressure, we also consider further fluid properties in the well-bore to express Δ_{ij}^p. This results in some additional unknowns that are relevant for the modeling of wells. For details, we refer to Jiang [52].

2.4.2.3 Effect on the Linear Systems

The well equations have to be taken into account in addition to the mass balances within the reservoir. That is, the Jacobian is extended by a well part and reads as:

$$J_{withWells} = \begin{pmatrix} J_{RR} & J_{RW} \\ J_{WR} & J_{WW} \end{pmatrix}. \tag{2.34}$$

The block J_{RR} is based on the mass balances within the reservoir and corresponds to (2.14). The part J_{WW} is resulting from the interaction of well cells with each

other. The further two blocks describe the coupling between reservoir and well. J_{RW} results from the source terms' dependence on the well pressures. J_{WR} describes how the reservoir influences the well. Just like J_{WW}, it depends on the model and the type of control. This makes general statements regarding matrix properties rather difficult.
We do not further discuss the concrete impacts on the matrix properties for two (related) reasons:

- The number of well equations, i.e., the dimension of the block J_{WW}, is typically very small compared to J_{RR} and the applicability of a solver method is mainly determined by the properties of the reservoir part.
- According to practical experience, the well equations are typically not critical for AMG. In the later chapters we will consider linear systems from industrial simulations. There, from AMG's perspective, the source terms may have a severe influence on the matrices. The well equations in all considered cases did not cause any difficulties.

For our further discussion on matrix properties, we assume all wells are BHP controlled and modeled by the standard well model with neglecting the compressibility within the well. Then J_{WW} vanishes.

Remark 2.33. *If there should be situations where the well equations themselves, i.e., J_{WW} and the respective couplings, cause problems for the linear solver, it is a reasonable remedy to switch to some decoupling approach, e.g., alternating Schwarz approaches. That is, the simulator iterates between reservoir part and well part.*
We will discuss solver approaches for the reservoir part in detail in the following chapters. We will also discuss how to deal with the influences from the source term.
The second Schwarz block only contains the mere well part J_{WW}. Due to the small size of J_{WW}, a direct solver is an option that can typically be realized efficiently. It reacts less sensitively to different matrix properties than multigrid. △

Remark 2.34. *In the context of multigrid, it also is a considerable option to not involve the well equations in the hierarchy, but use the hierarchy for the reservoir part, only.* △

2.4.3 Fracture Modeling

Fractures and faults within a geological structure result from seismic forces on the rock and are characterized by drastically higher porosities and permeabilities compared to the surrounding rock. Due to these drastic jumps of the material properties, it would be difficult to use the same discretization techniques that we have used for the non-fracture parts.
One option is to use special discretization techniques, as described by Matthäi et.al. [70]. Regarding the matrix properties, this may result in drastic jumps of the coefficients, however, the general properties that we have found so far are not influenced.

Another option is to separate the domain into fractures and non-fractures, as

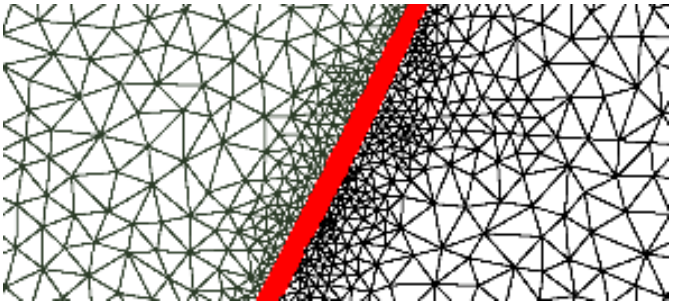


FIGURE 2.8: Discretization of two reservoir parts, divided by a fracture (red). Different methods are common to capture the flux between these different parts and the fracture.
Picture based on the CSMP software by Matthäi et.al.

sketched in Figure 2.8. For the rock, i.e., the non-fracture part, our standard discretization techniques can be used, while the fractures are treated differently. There are essentially two such approaches, the *dual porosity dual permeability formulation* and the *fracture network model*. Descriptions are, for instance, given by Chen et.al. or Sarma and Juliusson [21, 54, 89], respectively, and the references therein. Both formulations model the interaction between reservoir and fracture via source terms. The derivation is comparable to what we have discussed in the context of wells. Hence, we expect the same sort of influences on the linear systems.
The fluid flow within the fracture is modeled differently in both models. However, in neither case we have an influence on the linear systems that we are going to solve with an AMG approach. For details on the modeling of the flow in the fractures, we refer to the literature mentioned above.

2.5 Adaptive Implicit Simulations

While the fully implicit modeling approach from Section 2.3.2 has the advantage of being unconditionally stable for any time step size, it has the disadvantage of each time step being computationally more expensive than in the partly explicit approaches from Section 2.3.1. The larger time steps typically outweigh this disadvantage. However, with the *adaptive implicit modeling* (AIM) approach a compromise between fully implicit and partly explicit is possible. For a detailed description of the approach, which is frequently used in practice, we refer to the description by Forsyth and Sammon [42].

The idea is to treat the non-pressure unknowns explicitly wherever this is possible without negative impacts on the accuracy of the discretization. The decision on whether to do so for a cell i in time step n is based on this cell's Courant number from the previous time step. A description is, for instance, given by Coats [29].

We treat the saturations explicitly in those cells where the Courant number does not exceed a certain limit. In these cells, the previous time step's saturations are used to compute properties like relative permeabilities. This drastically simplifies the sub-matrices A_{ss} and A_{ps} to be block diagonal in the point-wise numbering. This can be exploited to accelerate the linear solution process, as we will see in Section 6.2. For the pressure related blocks A_{pp} and A_{sp}, this proceeding does not make a difference.

2.6 Compositional Simulations

The hydrocarbon phases (oil and gas) consist of various chemical components such as methane, octane, propane, etc. The concrete decomposition of these components has an impact on the fluid flow as well. This is described by compositional formulations.

Each of the hydrocarbon components c can be present in the oily and the gaseous phase at the same time. The *concentration* $X_{c,\alpha}$ describes the amount of phase α that consists of component c. The decomposition of c in oil and gas depends on pressure and temperature. This dependency is expressed by a so-called *flash relation*. Whether or not this flash relation is involved in the linear solution process, depends on the model formulation.

2.6.1 Natural Variable Formulation

Fluid properties are expressed in terms of naturally measurable quantities (e.g., phase densities, etc.). In particular, the decomposition of the fluid is described by the saturations and concentrations, which both are unknowns that we compute with Newton's method. That is, we can model the volume-balance, i.e., the entire pore volume is filled by the fluid, with simple constraints:

$$\sum_{k=1}^{n_{comp}} X_{k,\alpha} = 1, \tag{2.35}$$

which needs to hold in addition to the analog constraint for the saturation (2.3). Because we use the same natural variables and properties as in the previous Black-Oil section, we can essentially use the same formulation for the mass balance equations. However, as we are considering the mass of the entire component c, we need to take the gaseous and the oily mass into account at the same time. Hence, the respective mass balances (2.10) from the Black-Oil model need to be combined, weighted by the concentrations. The mass balance for component c reads as:

$$0 = \underbrace{\sum_{\alpha=oil,gas} X_{c,\alpha}\{\frac{\partial}{\partial t}\varphi\rho_\alpha S_\alpha\}}_{Accumulation\ Part} - \underbrace{\nabla \sum_{\alpha=oil,gas} X_{c,\alpha}\{\rho_\alpha K\lambda_\alpha(\nabla p_\alpha - g\nabla d)\}}_{Flux\ Part} + \underbrace{\sum_{\alpha=oil,gas} X_{c,\alpha}\{\rho_\alpha q_\alpha\}}_{Source\ Part}. \tag{2.36}$$

For the water phase there is no difference compared to the Black-Oil model and we use the mass balance (2.10). A further description is, for instance, given by Cao [18].

We have to consider one mass balance equation but two concentration unknowns per component and, also with with the constraints on the saturations and concentrations, the system is under-determined so-far. We have already mentioned that the concrete amount of component c being gaseous and oily, respectively, is obtained from an equilibrium relation, the flash relation. Various approaches of modeling these flashes are reported in the literature and new methods still are being developed. We refer to Naji [75] and the references therein for further details. We exemplarily continue with the approach using Wilson's relation:

$$\frac{X_{c,gas}}{X_{c,oil}} = F_c := \frac{p_{ref}}{p_{tot}} e^{c_{flash}(1-\frac{T_{ref}}{T})}. \tag{2.37}$$

We assume the temperature to be constant for the moment and get back to this aspect later on. $c_{flash} > 0$ is a constant.

With the mass balances, flashes and constraints, we can associate each unknown with an equation and compute a Jacobian. In order to describe the properties of this Jacobian, we distinguish between primary and secondary equations. Primary equations describe interactions of the fluid in one cell with neighboring cells. In the current context of isothermal models, only the mass balances are primary equations. We call the unknowns associated with them primary unknowns. In contrast to this, secondary equations describe local relations, like flashes or constraints. The unknowns associated with these equations are secondary unknowns. As the flash relations depend on, for instance, pressure and temperature, we cannot simply eliminate the respective constraints from the system, as we did with the saturation-summation constraint (2.3) in the Black-Oil case[4].
The Jacobian schematically reads as:

$$J_{NVF} = \begin{pmatrix} A_{pr,pr} & A_{pr,sc} \\ A_{sc,pr} & A_{sc,sc} \end{pmatrix}. \tag{2.38}$$

When discussing AMG approaches for such problems, we will see that we, in order to reduce the computational time, can decouple the secondary equations from the systems without negative impacts on the applicability of AMG. This will require some knowledge on the properties of the four blocks that we are going to discuss in the following. We assume the pressure p_α, the saturations S_β, S_γ and the gas concentrations $X_{c,gas}$ to be primary variables. We still choose α to be the oil phase and it does not matter whether gas is β or γ. The oil concentrations are the secondary unknowns.

Corollary 2.35. *The sub-block $A_{pr,pr}$ has the same properties as the Jacobian (2.14) in Black-Oil simulations.*

Proof. The derivatives of the mass balances w.r.t. concentrations are analog to those w.r.t. saturations. Hence, if we define A_{ss} to be the block that is related to saturations and primary concentrations, the dimension of this block increases compared to the Black-Oil case. Its properties, however, remain the same.

[4]We could incorporate the flash calculations in the mass balance equations, just as we did with the capillary pressure relations. However, depending on the concrete formulation for the flashes, this would make the mass balance equations being quite complex. Therefore, this is typically not done in practice, but the secondary equations are used.

For the pressure derivatives it does not make a difference whether the mass balance holds per phase or per component. □

Corollary 2.36. *As the secondary unknowns are concentrations, the sub-block $A_{pr,sc}$ has the same properties as A_{ss}, i.e., it consists of sub-blocks that are Z-matrices and is (block) diagonally dominant.* △

Lemma 2.37. *The sub-blocks of $A_{sc,pr}$ are diagonal matrices. If the flash calculation is performed by Wilson's relation (2.37), then all entries are non-positive.*

Proof. The diagonality of the sub-matrices follows from the fact that the secondary equations describe local relations. Constraints can directly be formulated such that they result in the correct sign. The pressure derivative of the discretized flash calculation for cell i in time step n is:

$$\begin{aligned}\frac{\partial}{\partial [p_\alpha]_i^n}([F_c]_i^n[X_{c,oil}]_i^n - [X_{c,gas}]_i^n) &= [X_{c,oil}]_i^n \frac{\partial}{\partial [p_\alpha]_i^n}\frac{p_{ref}}{[p_\alpha]_i^n} e^{c_{flash}(1-\frac{T_{ref}}{[T]_i^n})} \\ &= -\frac{[X_{c,oil}]_i^n}{[p_\alpha]_i^n}[F_c]_i^n \leq 0.\end{aligned}$$

The saturation derivatives are zero and for the gas concentrations, which are primary variables, we have

$$\frac{\partial}{\partial [X_{c,gas}]_i^n}([F_c]_i^n[X_{c,oil}]_i^n - [X_{c,gas}]_i^n) = -1.$$

□

Lemma 2.38. *The sub-blocks of $A_{sc,sc}$ are diagonal matrices. If the flash calculation is performed by Wilson's relation (2.37), then these diagonal matrices are non-negative. The diagonal of $A_{sc,sc}$ is positive.*

Proof. The diagonality follows as above. Regarding the derivatives of the discretized flash calculations w.r.t. the secondary oil concentrations we have:

$$\frac{\partial}{\partial [X_{c,oil}]_i^n}([F_c]_i^n[X_{c,oil}]_i^n - [X_{c,gas}]_i^n) = [F_c]_i^n > 0.$$

□

Remark 2.39. *There are various other flash calculation approaches that aim at capturing the phase transitions more accurately. All of them are strictly local*

relations. Because they aim at approximating the same effect, we can expect comparable influences on the properties of the linear systems of equations. We do not discuss the details of other flash calculations approaches in detail here.
Moreover, for more sophisticated flash calculations, the respective derivatives become increasingly complicated. We will see that the linear systems do not involve the flash relations with the volume balance formulation that we describe in the following section. △

Remark 2.40. *The formulation (2.36) for the mass balances assumes the velocity of each component in a phase to be equal, i.e.* $\vec{v}_{c,\alpha} = \vec{v}_\alpha$. *According to Wan [103], this is common in reservoir simulations and we will proceed this way. However, by taking so-called macroscopic diffusive-dispersive effects into account, the model could be made more precise with this regard. The flux part of the above mass balance then turns into*

$$-\nabla \sum_{\alpha=oil,gas} X_{c,\alpha}\{\rho_\alpha(K\lambda_\alpha(\nabla p_\alpha - g\nabla d) - D_{c,\alpha}\nabla X_{c,\alpha})\}. \tag{2.39}$$

Here $D_{c,\alpha}$ *is obtained from Fick's Law. We refer to Wan [103] for details. For our discussion it is sufficient to note that this modification does not have any impacts on the pressure sub-blocks, as* $D_{c,\alpha}$ *is independent of the pressure. The impacts on the saturation and concentration related matrix parts can be expected to be small. Due to both entities being involved in the accumulation term, which involves the time step, we can still expect the respective matrices to be strongly diagonally dominant.* △

2.6.2 Volume Balance Formulation

While the previously described natural variable formulation has the advantage of using the natural fluid properties, it has the disadvantage of requiring flash calculations being involved in the linear systems. This is the other way round for the *volume balance formulation*. It computes flashes independently of the linearization, which allows for more freedom in modeling them. From our perspective, this has the advantage of the linear solver not being concerned with flashes. However, the formulation expresses the mass balance for component c in terms of the respective number of moles, N_c, instead of the relative quantities saturation and concentration. For a hydrocarbon component c, the molar mass balance reads as

[31]:

$$0 = MB_c := \frac{\partial}{\partial t} M_c^{mol} N_c - X_{c,oil} \nabla T_{c,oil}^{mol} (\nabla p_{oil} - g \nabla d) - X_{c,gas} \nabla T_{c,gas}^{mol} (\nabla p_{gas} - g \nabla D) + q_c^{mol}. \tag{2.40}$$

Here M_c^{mol} is the molar mass of component c. Hence, $N_c M_c^{mol}$ gives the mass of all particles of component c. $T_{c,oil}^{mol}$ and $T_{c,gas}^{mol}$ are the molar transmissibilities of the oily and gaseous c, respectively. The computation is analog to what we had before (cf. (2.16)). The concentrations $X_{c,oil}$ and $X_{c,gas}$ are known from flash calculations. Finally, q_c^{mol} is the molar source term.
For the water phase we have an analog mass balance, where we treat the entire water as one component.

Because no saturations are considered anymore, the saturation constraint (2.3) cannot be used as volume balance[5]. In the fully implicit formulation Wong et.al. [109] showed that this can alternatively be modeled by the following relation, as also described by Collins et.al. [31]:

$$0 = VB := \sum_{k=1}^{n_{comp}} \frac{N_k M_k^{mol}}{\rho_k} - \varphi. \tag{2.41}$$

We associate the pressure unknown, p, with the volume balance VB and each number of moles, N_c, is associated with the respective mass balance MB_c. Schematically, we have the following Jacobian, computed from the discretized equations:

$$J_{VBF} = \begin{pmatrix} A_{VB,p} & A_{VB,N} \\ A_{MB,p} & A_{MB,N} \end{pmatrix}, \tag{2.42}$$

with the different sub-blocks as:

$$A_{VB,p} = \begin{pmatrix} \frac{\partial [VB]}{\partial [p]} \end{pmatrix} \qquad A_{VB,N} = \begin{pmatrix} \frac{\partial [VB]}{\partial [N_1]} & \cdots & \frac{\partial [VB]}{\partial [N_{comp}]} \end{pmatrix}$$
$$A_{MB,p} = \begin{pmatrix} \frac{\partial [MB_1]}{\partial [p]} \\ \vdots \\ \frac{\partial [MB_{n_{comp}}]}{\partial [p]} \end{pmatrix} \quad A_{MB,N} = \begin{pmatrix} \frac{\partial [MB_1]}{\partial [N_1]} & \cdots & \frac{\partial [MB_1]}{\partial [N_{n_{comp}}]} \\ \vdots & \ddots & \vdots \\ \frac{\partial [MB_{n_{comp}}]}{\partial [N_1]} & \cdots & \frac{\partial [MB_{n_{comp}}]}{\partial [N_{n_{comp}}]} \end{pmatrix}. \tag{2.43}$$

Just as with the Jacobian for the natural variable formulation, we can lead back the properties of the matrix blocks that result from derivatives of the mass balance

[5]This is the same situation as with the respective partly explicit method from Section 2.3.1.

equations to the Black-Oil case. We will exploit these properties when discussing a linear solver approach in Section 5.1.2.

Corollary 2.41. *Regarding the pressure dependence there is no difference between the above mass balance and the one from Black-Oil simulations. Hence, the sub-block $A_{MB,p}$ has the same properties as A_{sp} from Black-Oil simulations (cf. Corollary 2.22 and Theorem 2.25). That is, the sub-matrices are M-matrices under certain conditions for the time steps and external sources.* △

Corollary 2.42. *Analogously, the sub-block $A_{MB,N}$ has the same properties as A_{ss} from Black-Oil simulations (cf. Theorem 2.28). That is, the sub-blocks are Z-matrices and it is (block) diagonally dominant.* △

Lemma 2.43. *The sub-block $A_{VB,p}$ is a diagonal matrix and $A_{VB,p} \leq 0$.*

Proof. The diagonality follows from the fact that (2.41) per cell i only involves quantities of cell i. For cell i in time step n we have:

$$\frac{\partial}{\partial [p]_i^n}[VB]_i^n = \sum_{k=1}^{n_{comp}} [N_k M_k^{mol}]_i^n \frac{\partial}{\partial [p]_i^n} \frac{1}{[\rho_k]_i^n} - \frac{\partial}{\partial [p]_i^n}[\varphi]_i^n.$$

For the porosity there is $\frac{\partial}{\partial [p]_i^n}[\varphi]_i^n \geq 0$ and for the molar densities we also have $\frac{\partial}{\partial [p]_i^n}[\rho_k]_i^n \geq 0$. □

Lemma 2.44. *$A_{VB,N}$ has sub-blocks that are diagonal matrices with positive entries.*

Proof. The diagonality follows as above. Regarding the matrix entry that corresponds to one of the components c in cell i for time step n, we have:

$$\frac{\partial}{\partial [N_c]_i^n}[VB]_i^n = \frac{M_k^{mol}}{[\rho_c]_i^n} > 0.$$

□

2.7 Disappearance of Phases and Components

Up to now we have assumed all phases and components to be present in each cell at any time. We are now going to look at possible impacts from the disappearance of certain components or phases on the linear systems of equations.

2.7.1 Disappearance of Phases

If a simulator should allow for a phase, e.g., oil or water, to disappear[6], then, in the respective cells, there is just one unknown (the respective saturation) and one equation (the respective mass balance) removed from the system.

This is different for the gas phase. Depending on the pressure, it can disappear and reappear. Disappearing means it is compressed such that it dissolves with the oil phase. As long as there is gas present, we call the oil to be *saturated*. We have seen that taking some gas being dissolved into account does not influence the properties of the linear systems to be solved (cf. Remark 2.24). However, at a certain pressure all gas will be dissolved and the gas phase disappears. The oil is called *under-saturated* then. This critical pressure is called *bubble point pressure* p_{bp}, as gas bubbles fall out the oil phase at lower pressures. The gas' mass did not disappear, only the phase did. Hence, we still have to take the gas' mass balance (2.11) into account, with the gas-oil ratio R_{og} (cf. Remark 2.19) describing the dissolved gas. As there is no free gas under the current pressure, this mass balance turns into:

$$0 = \frac{\partial}{\partial t}\varphi R_{og} S_{oil}\rho_{oil} - \nabla(R_{og}\rho_{oil}\lambda_{oil}\vec{\vec{K}}(\nabla p_{oil} - g\nabla d)) + R_{og}\rho_{oil}q_{oil}. \tag{2.44}$$

The gas saturation S_{gas} is not unknown anymore, but zero. Following the discussion by Forsyth and Sammon [42], the bubble point pressure is associated with the gas mass balance, instead of S_{gas}. That is, the matrix block that resulted from the derivative of (2.11) w.r.t. S_{gas} in the saturated case, is replaced by a block that results from the derivative of (2.44) w.r.t. p_{bp} in the under-saturated case.
In the under-saturated case, the gas-oil ratio depends on p_{bp} (while it depends on p_{gas} in the saturated case). As R_{og} is involved in the flux part, we have negative contributions on the off-diagonals of the respective sub-matrix, and positive ones on the diagonal. Because R_{og} also is involved in the accumulation part, depending on the time step, we can expect the respective sub-matrix to be strongly diagonally dominant.
Therefore, although we replace a block of A_{ss} by a different one, the basic properties of A_{ss} do not change compared to the saturated case.

[6]We note that in reality there are always hydrocarbons and water present in any point at any time. The oil saturation will never fall below a positive oil residual saturation [36, 94]. Analogously, for the water an irreducible water saturation exists, where S_{water} will never fall below.

2.7.2 Disappearance of Components

In the compositional case not necessarily all components need to be present at any point of the reservoir, even if all phases are.
In the NVF-case, the respective concentrations in the oil and gas phase are zero and at the same time the respective mass balance and flash relation are obsolete. Hence, the Jacobian (2.38) involves less unknowns, but its properties remain.
In the VBF-case, there is one number of moles and one mass balance being obsolete. Hence, the Jacobian involves less unknowns, but with the same properties as with all components being present.

2.8 Thermal Simulations

So far we have neglected thermal influences on the production process by assuming a constant temperature. However, to extract as much oil and gas from a reservoir as possible, enhanced oil recovery (EOR) techniques involve the use of heat to make oil more flowable. This can be realized by the injection of hot steam instead of water, as illustrated in Figure 2.9, a method referred to as steamflooding. But also in-situ heating and even combustion processes are possible. In any case, thermal energy is exploited and we cannot assume the *temperature*[7] T to be constant. The temperature is directly related to the thermal energy, which needs to be conserved. Just as with the mass balances, the energy that is accumulated at a point over time can only result from energy flow to and from other points in the reservoir and from external sources. Both types of flow hereby have to consider energy flow itself and the fact that fluid flow also transports energy. The ability of a fluid to transport energy by its flow is described as the *enthalpy*[8] H_{tot}. The accumulated energy at a point is related to the *internal energy* U_{tot} of the fluid, or the rock (U_{rock}). Following Livescu et.al. [65], these entities are given as

$$H_{tot} = U_{tot} + \frac{p_{tot}}{\rho_{tot}}, \quad U_{tot} = C_{tot}(T - T_{ref}) \text{ and } U_{rock} = C_{rock}(T - T_{ref}), \quad (2.45)$$

where C_{tot} and C_{rock} are the constant heat capacities of fluid and rock. The ability of heat to flow through the rock in different directions is described by a *thermal*

[7] We only consider one temperature T for the entire system, which we should not mix up with the phase-wise transmissibility T_α.

[8] In our discussion, as in most descriptions, we only consider one temperature that holds for all phases. Hence, we use the total fluid properties in order to describe the energy balance.

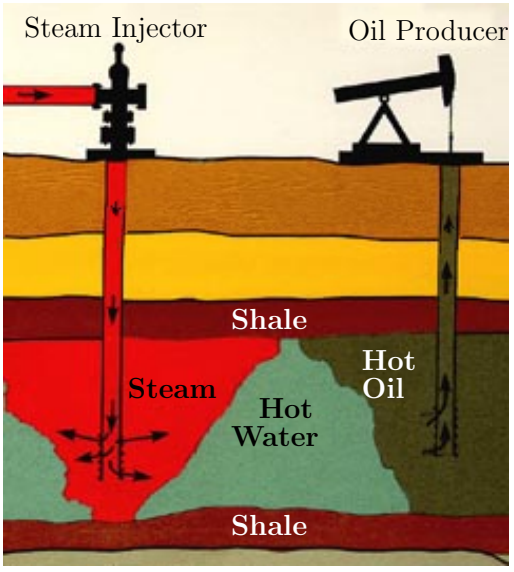


FIGURE 2.9: Steam instead of water is injected into the reservoir in order to reduce the oil's viscosity.
Picture from the US Department of Energy, via the Wikipedia (Steam_eor1.jpg)

permeability $\vec{K}_T$, or K_T for brevity. It is the analogue of the fluid permeability K. However, as heat may flow through rock, not only through pores, it typically drastically differs from K regarding the concrete values. We refer to the literature [73, 78, 86] for further details on the derivation of the energy balance. The energy conservation equation reads as:

$$0 = \underbrace{\frac{\partial}{\partial t}(\varphi\rho_{tot}U_{tot} + (1-\varphi)\rho_{rock}U_{rock})}_{\textit{Accumulation}} + \underbrace{\nabla\rho_{tot}H_{tot}\vec{v}_{tot}}_{\substack{\textit{Heat transport}\\ \textit{with fluid flux}}} - \underbrace{\nabla K_T \nabla T}_{\textit{Heat flux}} + \underbrace{q_T + \rho_{tot}H_{tot}q_{tot}}_{\substack{\textit{External sources,}\\ \textit{direct and via fluid}}}. \tag{2.46}$$

We can use Darcy's law (2.9) to compute the fluid velocity $\vec{v}_{tot}$. If we associate the energy balance with the temperature unknown, the further proceeding is just as in the isothermal models. We discretize the equation and then linearize the full set of equations by Newton's method. That is, compared to the Jacobians that we have from Black-Oil or compositional simulations, there is just one additional unknown and the respective block of equations. The sub-matrices that we already had in isothermal simulations remain unchanged. We will discuss the compositional case in Section 2.8.4 and consider Black-Oil problems for the moment. The extended

Jacobian reads as:

$$J_{thermal} = \begin{pmatrix} A_{pp} & A_{ps} & A_{pT} \\ A_{sp} & A_{ss} & A_{sT} \\ A_{Tp} & A_{Ts} & A_{TT} \end{pmatrix}. \tag{2.47}$$

Let us consider the one dimensional discretization grid from Section 2.3.2.1. Then, regarding cell i and time step n, the discrete energy balance, we denote it as $[E]_i^n$, reads as:

$$\begin{aligned}
[E]_i^n = \quad & \left.\begin{aligned} &\tfrac{1}{\Delta t}([\varphi\rho_{tot}U_{tot} + (1-\varphi)\rho_{rock}U_{rock}]_i^n \\ &\quad - [\varphi\rho_{tot}U_{tot} + (1-\varphi)\rho_{rock}U_{rock}]_i^{n-1}) \end{aligned}\right\} =: [E^{Accu}]_i^n \\
& \left.\begin{aligned} &-\tfrac{1}{hA}([T_{tot}H_{tot}]_{i-\frac{1}{2}}^n([p_{tot}]_{i-1}^n - [p_{tot}]_i^n) \\ &\quad + [T_{tot}H_{tot}]_{i+\frac{1}{2}}^n([p_{tot}]_{i+1}^n - [p_{tot}]_i^n)) \end{aligned}\right\} =: [E^{FluidFlux}]_i^n \\
& \left.\begin{aligned} &-\tfrac{1}{hA}([K_T]_{i-\frac{1}{2}}([T]_{i-1}^n - [T]_i^n) + [K_T]_{i+\frac{1}{2}}([T]_{i+1}^n - [T]_i^n)) \end{aligned}\right\} =: [E^{HeatFlux}]_i^n \\
& \left.\begin{aligned} &+[q_T]_i^n + [\rho_{tot}H_{tot}q_{tot}]_i^n \end{aligned}\right\} =: [E^{Source}]_i^n.
\end{aligned} \tag{2.48}$$

We denote the different parts of the equation as $[E^{Accu}]$, etc. (or $[E^{Accu}]_i^n$, if we refer to a particular cell and time step). In higher dimensions, the discretization would work analogously to what we have discussed in Section 2.3.2.2. That is, if MPFA schemes were used, the discretization of the fluid flux term might change. However, our later result that A_{Tp} has the same properties as A_{pp}, still holds.
By computing the temperature, pressure and saturation derivatives of the discrete energy balance, as well as the temperature derivatives of the discrete mass balances, we obtain the new sub-blocks of the thermal Jacobian. We are going to discuss the matrix properties in the following.

2.8.1 Matrix Properties of the Temperature Part

Regarding the temperature related matrix $A_{TT} := \frac{\partial}{\partial[T]}[E]$, we have to consider the derivatives of the different summands of (2.48) w.r.t. the temperature in a cell i and the neighboring cells. We define $A_{TT} := A_{Eaccu,T} + A_{Efluidflux,T} + A_{Eheatflux,T} + A_{Esource,T}$ with the four different matrices describing the influence of the respective summands of the discrete energy balance.
As with the fluid flux, $A_{Eaccu,T} := \frac{\partial}{\partial[T]}[E^{Accu}]$ is a diagonal matrix. With the

internal energy (2.45), we have:

$$\begin{aligned}\frac{\partial}{\partial[T]_i^n}[E^{Accu}]_i^n =& \frac{\varphi}{\Delta t}([U_{tot}]_i^n \frac{\partial}{\partial[T]_i^n}[\rho_{tot}]_i^n + [\rho_{tot}]_i^n[C_{tot}]_i^n) \\ &+ \frac{1-\varphi}{\Delta t}([U_{rock}]_i^n \frac{\partial}{\partial[T]_i^n}[\rho_{rock}]_i^n + [\rho_{rock}]_i^n[C_{rock}]_i^n).\end{aligned} \tag{2.49}$$

The temperature derivative of the density is negative. However, because the temperature dependence of the density typically is small compared to the one of the internal energy, and $C_{tot}, C_{rock} > 0$, we can expect $A_{Eaccu,T} \geq 0$ in practical simulations.

Let us continue with $A_{Efluidflux,T} := \frac{\partial}{\partial[T]}[E^{FluidFlux}]$. Regarding cell i, for the derivative w.r.t. the temperature in this cell, we have:

$$\begin{aligned}&\frac{\partial}{\partial[T]_i^n}[E^{FluidFlux}]_i^n = \\ &\quad \frac{1}{hA}(([p_{tot}]_i^n - [p_{tot}]_{i-1}^n)([H_{tot}]_{i-\frac{1}{2}}^n \frac{\partial}{\partial[T]_i^n}[T_{tot}]_{i-\frac{1}{2}}^n + [T_{tot}]_{i-\frac{1}{2}}^n \frac{\partial}{\partial[T]_i^n}[H_{tot}]_{i-\frac{1}{2}}^n) \\ &\quad + ([p_{tot}]_i^n - [p_{tot}]_{i+1}^n)([H_{tot}]_{i+\frac{1}{2}}^n \frac{\partial}{\partial[T]_i^n}[T_{tot}]_{i+\frac{1}{2}}^n + [T_{tot}]_{i+\frac{1}{2}}^n \frac{\partial}{\partial[T]_i^n}[H_{tot}]_{i+\frac{1}{2}}^n)).\end{aligned} \tag{2.50}$$

The derivatives w.r.t. the temperatures in the neighboring cells $i \pm 1$ are computed analogously.

We remember the transmissibilities (2.16) have been discretized with upstream weighting (2.17). Analogously to the argumentation regarding the saturation derivatives (2.23) and (2.24) of the flux part of the discrete mass balances, we conclude that the temperature derivative will vanish where the pressure differences $([p_{tot}]_i^n - [p_{tot}]_{i\pm1}^n)$ in (2.50) are negative. However, we still cannot rigorously determine the signs of the off-diagonals and diagonals of $A_{Efluidflux,T}$:

- For the enthalpy (2.45) we have to consider the internal energy's temperature derivative, which is positive. As the density's temperature derivative is negative, the temperature derivative of the enthalpy is positive.
- For the transmissibility we have to consider the temperature derivative of the term $\frac{\rho}{\mu}$. Because the temperature derivative of the viscosity is negative, we have a positive derivative of $\frac{1}{\mu}$. However, the temperature derivative of the density, although typically small, is negative.

Hence, enthalpy and viscosity give a positive contribution to the matrix' diagonal and negative contributions to the off-diagonals, whereas this is the other way round for the density in the transmissibility. However, the latter influence typically does not dominate.
For $A_{Eenergyflux,T} := \frac{\partial}{\partial[T]}[E^{HeatFlux}]$ we have:

$$\frac{\partial}{\partial[T]_i^n}[E^{HeatFlux}]_i^n = \frac{[K_T]_{i-\frac{1}{2}} + [K_T]_{i+\frac{1}{2}}}{hA} \text{ and } \frac{\partial}{\partial[T]_{i\pm1}^n}[E^{HeatFlux}]_i^n = \frac{-[K_T]_{i\pm\frac{1}{2}}}{hA}. \tag{2.51}$$

We assume the thermal permeability is non-zero between any two adjacent cells and remember that we had Dirichlet boundary conditions for the temperature. Then $A_{Eenergyflux,T}$ is an spd M-matrix.
Finally, we note that the source terms are local effects and $A_{Esource,T} := \frac{\partial}{\partial[T]}[E^{Source}]$ is diagonal. Just as with the fluid source term (cf. Section 2.4.1.2), we find the temperature derivative of the thermal source term q_T to be positive. However, the influence from the energy transport by fluid flux is unpredictable. For an injection well q_{tot} is negative and we have the negative density derivative w.r.t. temperature. Both are combined with the positive derivative of the enthalpy.

In a typical thermal simulation, the matrix part that results from energy flux dominates the one from the fluid's energy transport. We conclude that $A_{Efluidflux,T} + A_{EenergyfluxT}$ has a positive diagonal and non-positive off-diagonals.
In addition, the internal energy's dependence on the temperature typically is much stronger than the one of the density. Therefore, $A_{Eaccu,T} + A_{Esource,T}$ is a positive diagonal matrix, where the time step size has a direct impact on the size of the diagonal entries. From both we conclude:

Corollary 2.45. *In a typical thermal simulation, A_{TT} is an M-matrix.* △

Remark 2.46. *The diagonal dominance of A_{TT} may be higher or lower, depending on whether the convective and, especially, accumulative influences dominate the diffusive ones. This may also vary between different cells.* △

2.8.2 Matrix Properties of the Temperature-Flux Coupling

Regarding $A_{Ts} := \frac{\partial}{\partial[S]}[E]$ we note that neither the thermal quantities internal energy and enthalpy (2.45) themselves, nor the energy flux and source term are saturation dependent. However, we still have to consider the saturation in the

total internal energy U_{tot}, which is the sum over the phases, weighted by the saturations. This gives positive contributions to the diagonal. Besides this, we have the saturation derivatives of the energy transport from fluid flux, which are of the same structure as the ones from fluid flux itself. Therefore, we can state:

Corollary 2.47. *Each block in A_{Ts} has the same properties as a diagonal block in A_{ss}. It is a diagonally dominant M-matrix.* △

The situation is more involved for the pressure related part $A_{Tp} := \frac{\partial}{\partial[p_\alpha]}[E]$. Let us discuss the pressure derivatives of the different parts of the discrete energy balance (2.48) in order. Regarding the accumulation part, only[9] the fluid density is pressure dependent, but it effects the diagonal only. We have

$$\frac{\partial}{\partial[p_\alpha]_i^n}[E^{Accu}]_i^n = \frac{1}{\Delta t}[\varphi U_{tot}]_i^n \frac{\partial}{\partial[p_\alpha]_i^n}\rho_{tot} > 0. \tag{2.52}$$

For the pressure derivative of the energy transport from fluid flux, exemplarily for the local influence (i.e., the contribution to the diagonal), we have

$$\frac{\partial}{\partial[p_\alpha]_i^n}\{-\frac{1}{hA}([T_{tot}H_{tot}]_{i-\frac{1}{2}}^n([p_{tot}]_{i-1}^n - [p_{tot}]_i^n) + [T_{tot}H_{tot}]_{i+\frac{1}{2}}^n([p_{tot}]_{i+1}^n - [p_{tot}]_i^n))\}. \tag{2.53}$$

In Section 2.3.2.1 we have already discussed that the pressure derivative of the transmissibility is positive. Also the pressure derivative of the enthalpy (2.45) is positive and so is the one of $[T_{tot}H_{tot}]_{i\pm\frac{1}{2}}^n$. Our argumentation regarding the properties of $A_{flux,p}$ from Section 2.3.2.1 can be applied analogously here. We conclude that the influence of the energy transport from fluid flux results in a, not necessarily diagonally dominant, Z-matrix.

The energy flux term is not pressure dependent. This also holds for the energy source q_T.

We finally have to consider the pressure dependence of the term resulting from energy transport with fluid sources:

$$\frac{\partial}{\partial[p_\alpha]_i^n}[\rho_{tot}H_{tot}q_{tot}]_i^n. \tag{2.54}$$

As the pressure derivative of $[\rho_{tot}H_{tot}]_i^n$ is positive, we have the same properties as in $A_{source,p}$ (cf. Section 2.4).

Altogether this yields:

[9] We will get back to mechanical effects on the porosity later. With the simplified pressure dependence of the porosity from Remark 2.18, the result regarding A_{Tp} is the same.

Corollary 2.48. *A_{Tp} has the same properties as A_{pp}. That is, under certain conditions regarding the fluid sources and the time step size, it is an M-matrix (cf. Theorem 2.25).* △.

2.8.3 Matrix Properties of the Flux-Temperature Coupling

Regarding $A_{pT} := \frac{\partial}{\partial[T]}[M_\alpha]$ and $A_{sT} := \frac{\partial}{\partial[T]}[M_{\beta,\gamma}]$ we note that both have comparable structures, because both result from the derivative of discrete mass balances (2.18) w.r.t. temperature. These equations are temperature dependent in the densities and mobilities. Both have negative derivatives w.r.t. the temperature, as we have discussed in Section 2.8.1.

Lemma 2.49. *Assuming no injection wells to be present, then $\exists \Delta t_{max} : \forall \Delta t \leq \Delta t_{max} : -A_{pT}$ is an M-matrix. With injection wells we have the same implications as with A_{pp}.*

Proof. The negative derivatives of density and mobility w.r.t. the temperature give a negative derivative of the transmissibility (2.16). Analogously to the discussion on the temperature derivative of the energy transport by fluid flux (2.50), we can expect the upstream weighting for the transmissibility to ensure that the resulting off-diagonal entries are non-negative, while the diagonal ones are negative.
The influence of the accumulation part results from $\frac{1}{\Delta t}[\varphi S_\alpha]_i^n \frac{\partial}{\partial[T]_i^n}[\rho_\alpha]_i^n$, which is negative. As long as there are no injection wells, the same holds for the temperature derivative of the source part $[q_\alpha]_i^n \frac{\partial}{\partial[T]_i^n}[\rho_\alpha]_i^n$.
The above considerations allow to apply the argumentation from Theorem 2.25, which proves the lemma. □

Because the accumulation part influences the diagonal only, depending on the concrete time step size, we can even expect A_{pT} and A_{sT} to be quite diagonally dominant.

2.8.4 Matrix Properties in Compositional and Thermal Simulations

So far, we have seen that the Jacobian in a thermal Black-Oil model is an extension of the isothermal version. We will now see that the same holds for compositional simulations as well.

2.8.4.1 Natural Variable Formulation

In a NVF approach from Section 2.6.1, we consider the temperature to be a primary unknown. The above considerations then carry over to the matrix sub-block $A_{pr,pr}$ (cf. Equation (2.38)). We need to discuss the blocks $A_{pr,sc}$ and $A_{sc,pr}$:

- $A_{pr,sc}$ results from derivatives of the discrete energy balance (2.48) w.r.t. the secondary unknowns. Only concentrations and saturations are secondary unknowns. Hence, we can expect the same matrix properties as for A_{Ts}, i.e. Z-matrices. These are the same properties, as for the blocks of $A_{pr,sc}$ in the isothermal case (cf. Corollary 2.36).
- $A_{sc,pr}$ results from the temperature derivative of the discretized secondary equations. As the secondary equations only describe local relations, $A_{sc,pr}$ still consists of diagonal sub-blocks. As before, we limit our discussion to Wilson's relation (2.37) and have

$$\frac{\partial}{\partial [T]_i^n}([F_c]_i^n [X_{c,oil}]_i^n - [X_{c,gas}]_i^n) = [X_{c,oil}]_i^n \frac{\partial}{\partial [T]_i^n} \frac{p_{ref}}{[p]_i^n} e^{c_{flash}(1-\frac{T_{ref}}{[T]_i^n})} \leq 0.$$

  In Lemma 2.37 we have seen that the blocks of $A_{sc,pr}$ in the isothermal case to be diagonal and non-positive. Hence, we have the same properties in the isothermal and thermal case.

Due to the temperature unknown being a primary unknown and the energy balance equation being a primary equation, there are no impacts on $A_{sc,sc}$.

2.8.4.2 Volume Balance Formulation

In the VBF approach we have to express the energy balance (2.46) with the molar density of the total fluid, rather than the total density ρ_{tot}. This, however, does

not have impacts on the matrix properties. The Jacobian, compared to (2.42), is extended by the temperature unknown. If we denote the discrete energy balance as EB, we schematically have

$$J_{VBFthermal} = \begin{pmatrix} A_{VB,p} & A_{VB,N} & A_{VB,T} \\ A_{MB,p} & A_{MB,N} & A_{MB,T} \\ A_{EB,p} & A_{EB,N} & A_{EB,T} \end{pmatrix}. \tag{2.55}$$

- $A_{EB,T}$ is computed just as A_{TT} in the Black-Oil case.
- The same holds for $A_{EB,p}$ and $A_{EB,N}$, which are computed analogously to A_{Tp} and A_{Ts}, respectively.
- The dependency of the molar density on the temperature is comparable to the one of the phase density. Hence, $A_{MB,T}$ is computed just as A_{sT}.
- To obtain $A_{VB,T}$, we compute

$$\frac{\partial}{\partial T} \sum_{k=1}^{n_{comp}} \frac{N_k M_k^{mol}}{\rho_k} - \varphi = \sum_{k=1}^{n_{comp}} N_k M_k^{mol} \frac{\partial}{\partial T} \frac{1}{\rho_k} \geq 0.$$

  Hence, $A_{EB,T}$ is diagonal with non-negative entries, just as the blocks from $A_{VB,N}$ (cf. Lemma 2.44).

2.9 Geomechanics

The sophisticated production processes in today's reservoirs require to also consider geomechanical impacts. Injection of a fluid at further increasing pressures results in mechanical forces that effect the sub-surface rock structures. On the one hand, this can mean that existing pores are compacted and shifted, which results in changes of porosity and permeability. On the other hand, it can also result in fractures being created or widened, so-called *fracking*.

We are dealing with a fluid flow problem and a mechanical one at the same time. The fluid flow is modeled as we have described so far. Some material properties, especially the porosity, just may be subject to mechanical influences now.
In our applications, for the mechanical part we can use linear elasticity formulations, which are known from various other applications of structural mechanics. We refer to Marsden and Hughes [69] for a detailed introduction. What we need to discuss in the following is how the mechanical and the fluid flow problem are

linked. Our discussion on these topics will essentially follow Wan, Kim and Kim et.al. [55, 56, 103]. For a detailed introduction into poroelasticity models, we also refer to Coussy [32].

2.9.1 Mechanical Sub-Problem

Before discussing the way how both models are linked, let us briefly review the mechanical sub-problem and the mechanical properties. We follow Dung and Kim [36, 55], who also provide more details.

2.9.1.1 Mechanical Properties

Essentially, we have to distinguish two different, yet related quantities: stresses and strains. *Stress* is a working force per area, which we describe by a tensor $\vec{\vec{\sigma}}$, or σ for brevity. Due to the conservation of angular momentum, σ is symmetric. *Strain* describes the change in volume of a rigid body that results from stress. We describe it by the tensor $\vec{\vec{\epsilon}}$, or simply ϵ. When neglecting the deformation in different directions, but only considering the overall volume deformation, it is common to consider the volumetric strain $\epsilon_v := trace(\epsilon)$.
The strain results in a *mechanical displacement* $\vec{u} = \begin{pmatrix} u_x & u_y & u_z \end{pmatrix}^T = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix}^T$ for the rock in the different coordinate directions. Under the assumptions of the infinitesimal strain theory, i.e., assuming the deformation of rock is small compared to the volume of the rock, we can express the strain in terms of displacement as

$$\epsilon_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}), \tag{2.56}$$

with the coordinate directions x_k. For the volumetric strain this implies $\epsilon_v = \nabla u$. Besides the mechanical quantities of stress and strain, we need four material properties.

- *Poisson's ratio* ν relates strain resulting from the working direction of the acting force to strains from other directions. The bigger this value, the more a material will 'escape' in other directions, as it is compressed from one direction, i.e., the more incompressible it is in a mechanical sense.
- *Young's modulus* (also known as elastic modulus) $E := \frac{\sigma}{\epsilon}$ describes a relation for stress and strain. This describes a material's stiffness.

- *Biot's effective stress coefficient* α_{Biot} and *Biot's modulus* M_{Biot}, which characterize the interaction between particular solids in fluids. Both are positive quantities.

2.9.1.2 Computation of Strain by Linear Elasticity

Linear elasticity formulations relate the *effective stress* σ' to the strain and, hence, the displacements. In the next section we will describe how the effective stress depends on the fluid pressure and the overall stress σ. If we assume σ' to be given for the moment, the Lamé Equations give the relation to be fulfilled for the displacements. We refer to Marsden and Hughes [69] for further information on these equations that essentially generalize Hooke's Law. They read as:

$$-\nabla\sigma' = -\nabla\hat{\lambda}\nabla u - 2\nabla\hat{\mu}\epsilon_v, \tag{2.57}$$

with $\hat{\lambda} := \frac{2\hat{\mu}\nu}{1-2\nu}$ and $\hat{\mu} := \frac{E}{2(1+\nu)}$ the first and second Lamé constant, respectively. Both constants are positive.

We have to define boundary conditions for the mechanical problem to complete the model. Because the reservoir is completely located underneath the earth's surface, there is no free mechanical expansion. This induces Dirichlet boundary conditions for the linear elasticity problem (2.57). Only if effects up to the surface are considered, Neumann boundary conditions may be present there. We refer to Wan [103] for further discussions on boundary conditions.

2.9.2 Influence of the Fluid Flow on the Mechanical Forces

The effective stress from the above Lamé Equations is induced by the fluid pressure in the reservoir. Our discussion follows Wan, Kim and Kim et.al. [55, 56, 103], who also provide further details. The effective stress is described by Terzaghi's effective stress relation, which is also known as first fundamental equation of poroelasticity.

It relates the stress σ with the fluid pressure[10]:

$$\sigma' = \sigma + \alpha_{Biot} p_{tot} Id. \tag{2.58}$$

For the overall stress σ, the momentum needs to be conserved. We can expect the gravity to act as main body force in a reservoir. Then the conservation of momentum reads as:

$$0 = \nabla\sigma + \rho_b g. \tag{2.59}$$

ρ_b is the bulk density, the combination of fluid and rock density that is weighted by the porosity. g is the gravitational constant.
By combining the previous two relations, we can express the effective stress gradient that we used in the Lamé Equations as:

$$\nabla\sigma' = \alpha_{Biot} \nabla p_{tot} Id - \rho_b g. \tag{2.60}$$

Remark 2.50. *If thermal expansion shall be incorporated in the model, following Settari and Walters [92], the effective stress gradient reads as:*

$$\nabla\sigma' = \alpha_{Biot} \nabla p_{tot} Id + \frac{E}{1-2\nu} \alpha_T \nabla T Id - \rho_b g. \tag{2.61}$$

Here $\alpha_T \geq 0$ is a constant coefficient. △

2.9.3 Influence of Mechanical Forces on the Fluid Flow

The way how the mechanical problem influences the fluid flow problem depends on which mechanical effects shall be modeled. Our discussion on linear solver approaches in this thesis is limited to geomechanical influences on the rock's porosity φ that is described by poroelasticity models. We will discuss this aspect first and only outline further mechanical influences on the fluid flow.

[10] In the general case, i.e., under so-called undrained conditions, we cannot use the total fluid pressure directly, but need to compute a bulk pressure p_b. Following Kim [55], the bulk pressure is related to the total fluid pressure as $p_b = p_{tot} - \alpha_{Biot} M_{Biot} \epsilon_v$. The Biot coefficient and the Biot modulus are constant. The volumetric strain is typically computed based on the previous time step. Hence, this does in principal not change the properties of the Jacobian and we do not further consider undrained conditions.

2.9.3.1 Impact on the Porosity

A change of the pore volume is a first mechanical effect that might already be induced by moderate pressures.

2.9.3.1.1 Compressible Rock

If we assume fluid pressures to compress the rock particles, then the fluid pressure has a direct impact on the porosity φ. These effects are considered in poroelasticity models. For a detailed description we refer to Coussy, Detournay and Cheng and Kim et.al. [32, 34, 56].

We have to take the non-constant porosity into account in the expression of the mass of a fluid phase (or component) that is accumulated in an arbitrary control volume over time (cf. the accumulation part of the mass balance (2.10)). For a single phase flow this is achieved with the second fundamental equation of poroelasticity. It gives the volume of a fluid that can be contained in a control volume of the reservoir, i.e., it expresses the same as the porosity does:

$$V_{tot} = \frac{m_{tot}}{\rho_{tot}} = \frac{1}{M_{Biot}} p_{tot} + \alpha_{Biot}\epsilon_v. \tag{2.62}$$

In the multi-phase case only a fraction of the pore volume is occupied by each phase. This fraction is described by the respective saturation. In the compositional case we would also need to involve the concentrations, but we leave it with the Black-Oil case here.

The mass balance from (2.10) then turns into

$$0 = \frac{\partial}{\partial t}\rho_\alpha S_\alpha \left(\frac{1}{M_{Biot}} p_{tot} + \alpha_{Biot}\epsilon_v\right) - \nabla \rho_\alpha \lambda_\alpha \vec{\vec{K}}(\nabla p_\alpha - g\nabla D) + \rho_\alpha q_\alpha. \tag{2.63}$$

2.9.3.1.2 Incompressible Rock

We can assume changes in the porosity not to result from a compression of rock particles, but only from a shift of them. This allows to directly express the porosity φ in terms of the volumetric strain, and therefore the displacements. We do not give the derivation of the model here and refer to Wan [103] for further discussion. Essentially, the model assumes the rock particles to be moving as a consequence of the mechanical forces. The mass conservation for the rock particles in combination

with the rock's velocity is then used to express the porosity as:

$$\varphi = 1 - (1 - \varphi_{ref})e^{-\epsilon_v + \epsilon_{ref}}. \tag{2.64}$$

Remark 2.51. *By only taking vertical strains into account, i.e., mechanical influences induced by gravitational forces, the Lamé equations (2.57) simplify such that (2.64) turns into the direct pressure dependence (2.7) from Remark 2.18:* $\varphi = \varphi_{ref} e^{C_{rock}(p_{total} - p_{ref})}$. *Further details are given by Dung [36].* △

2.9.3.2 Impact on the Permeability and Fracture Propagation

Not only the available pore space can be subject to mechanical influences, but also the connectivity of pores. On the one hand, the permeability tensor K may change under mechanical influences. There are various approaches described in the literature on how to incorporate this fact in the model. We will not discuss this in detail here, but refer to Dung [36] (Table 4-1), where different approaches of computing K from the stress σ are listed.
On the other hand, strong mechanical forces can open new, or extent existing fractures. This is also known as hydraulic fracturing, or fracking. A detailed description of modeling this effect is given by Philip et.al. [82]. Essentially, the speed $v_{fracture}$ that a fracture is growing, or propagated with, can be computed from the stress as

$$v_{fracture} = v_{max}(\sigma c_{fracture})^s. \tag{2.65}$$

This also is denoted as sub-critical crack-growth relation, where $c_{fracture}, s \geq 0$ are constants. The stress can be computed from the strain, i.e., from the displacements, as we have discussed earlier.

2.9.4 Properties of the Linear System

We only discuss effects of poroelasticity (cf. Section 2.9.3.1.1), as we will be concerned with linear systems from such simulations later on. As we have outlined initially, we are interlinking a mechanical and a fluid flow problem. That is, our overall linear systems can schematically be expressed as:

$$\begin{pmatrix} J_{F,F} & J_{F,M} \\ J_{M,F} & J_{M,M} \end{pmatrix} \begin{pmatrix} x_F \\ x_M \end{pmatrix} = \begin{pmatrix} f_F \\ f_M \end{pmatrix}. \tag{2.66}$$

The fluid unknowns x_F are the pressure, saturations and, depending of the type of model, concentrations or temperature. The fluid flow problem $J_{F,F}$ in principle is obtained as in the previous sections. The only difference is the pressure derivative of the mass balance (2.63) that takes a varying porosity into account. In contrast to the initial mass balance (2.10), the accumulation part of this equation involves the total pressure. Hence, if A_{pp}^{noMech} was the pressure sub-block from a simulation that did not consider mechanical effects, with some diagonal matrix P_M we have:

$$A_{pp} = A_{pp}^{noMech} + \frac{1}{\Delta t} P_M. \tag{2.67}$$

P_M is diagonal because the accumulation part of the discretized mass balance per cell i still only involves properties from this cell. As $M_{Biot} > 0$, we have $P_M > 0$. Therefore, all previous results on the matrix properties regarding A_{pp} persist. Adding positive values to the diagonal can be expected to be beneficial for any of the linear solver approaches that we are going to discuss in the following chapters. The same holds for the blocks of A_{sp}. The properties of the saturation (or concentration) related blocks do not change compared to the models that did not consider geomechanical effects.
The mechanical unknowns are the displacements u and the mechanical sub-problem $J_{M,M}$ results from the discretization of the Lamé Equations (2.57). For such problems by themselves, algebraic multigrid approaches are known to be an option to consider and we do not further discuss the matrix properties.

2.9.4.1 Black-Oil Flow Model

We first consider the fluid flow to be described by a Black-Oil model. For the two blocks that couple flow and mechanics, we first consider $J_{M,F}$. It results from the gradient of the total pressure that was involved in the description of the effective stress (2.60). In our formulation of the Lamé Equations (2.57), a multiplication with -1 is involved. The total pressure (cf. (2.4)) depends on pressure and saturations. Hence, each block of $J_{M,F}$ results from the discretization of a gradient, multiplied with $-\alpha_{Biot}$.
The coupling between fluid flow and mechanics, $J_{F,M}$, in our situation is resulting from the volumetric strain. This is the divergence of the displacements in the accumulation parts of the mass balances (2.63). It involves a multiplication with α_{Biot} and, in the discretized form of the mass balances, with $\frac{1}{\Delta t}$. Hence, we have

$\frac{-1}{\Delta t} J_{F,M} = J_{M,F}^T$.
Consequently, following Bergamaschi et.al. [12], with a coupling matrix $J_{M,F}$, and by scaling the flow part with Δt, we have

$$\begin{pmatrix} \Delta t J_{F,F} & -J_{M,F}^T \\ J_{M,F} & J_{M,M} \end{pmatrix} \begin{pmatrix} x_F \\ x_M \end{pmatrix} = \begin{pmatrix} \Delta t f_F \\ f_M \end{pmatrix}. \tag{2.68}$$

2.9.4.2 Compositional and Thermal Flow Models

In the scope of this thesis we will only be concerned with Black-Oil problems that are coupled with geomechanical models. However, for completeness we outline the following:

- Compositional, Natural Variable Formulation:
  The displacements are primary unknowns and for their coupling to and from pressure and saturations there is no change.
  Regarding the coupling between flux and mechanics we note that the pressure for each component in a particular phase is equal to the respective phase's pressure. Hence, we can equally well compute the total pressure as $p_{tot} = \sum_{\alpha,c} X_{c,\alpha} S_\alpha p_\alpha$. In this sense, regarding the coupling between mechanics and flow, we can treat concentrations analogously to saturations. We can limit this proceeding to the primary concentrations and the displacements are not coupled to any secondary unknown. The secondary equations (flash relations, etc.) are independent of mechanical influences.
- Compositional, Volume Balance Formulation:
  The volume consistency has to take the varying porosity into account. That is, (2.41) is replaced by

  $$\sum_{k=1}^{n_{comp}} \frac{N_k M_k^{mol}}{\rho_k} - (\frac{1}{M_{Biot}} p_{tot} + \alpha_{Biot} \epsilon_v) = 0. \tag{2.69}$$

  If we associated this equation with the total pressure, this was the only fluid related unknown that is coupled to the mechanics, and vice versa. The one coupling still is the transpose of the counter one, multiplied with -1. It does not even require a scaling by Δt. △
- Thermal:
  Following Remark 2.50, the mechanical equations now involve an additional

term that takes thermal expansion into account. This involves a temperature gradient that in our formulation of the Lamé Equations (2.57) is multiplied with -1. At the same time, the energy balance equation (2.46) is adjusted in the same way as the mass balance equations (2.63) have been. Hence, there still was $\frac{-1}{\Delta t} J_{F,M} = J_{M,F}^T$.

2.10 Summary

In the previous discussions we have investigated the properties of the linear systems that result from different types of reservoir simulation models. We have seen that more complex models are an extension of those models that involve less physical effects. This analogously holds for the linear systems resulting from the respective simulations. Figure 2.10 visualizes this 'evolution', and summarizes the properties of the different matrix blocks. It is also possible to omit some steps (e.g., thermal Black-Oil simulations, which means that the geomechanical and compositional parts are omitted). The figure does not show all these possible combinations, they follow in a straight forward way.

The depicted matrix properties have to be seen as 'base case' under moderate assumptions. Depending on time step size and further physical effects, there might be impacts on the linear systems. We should especially mention external sources and sinks (cf. Section 2.4). Although these only affect the matrix rows that correspond to those cells that are perforated, the effect on the matrix properties can be significant. We will discuss impacts on the linear solver, as well as ways to nevertheless handle the linear systems, in Chapter 4.

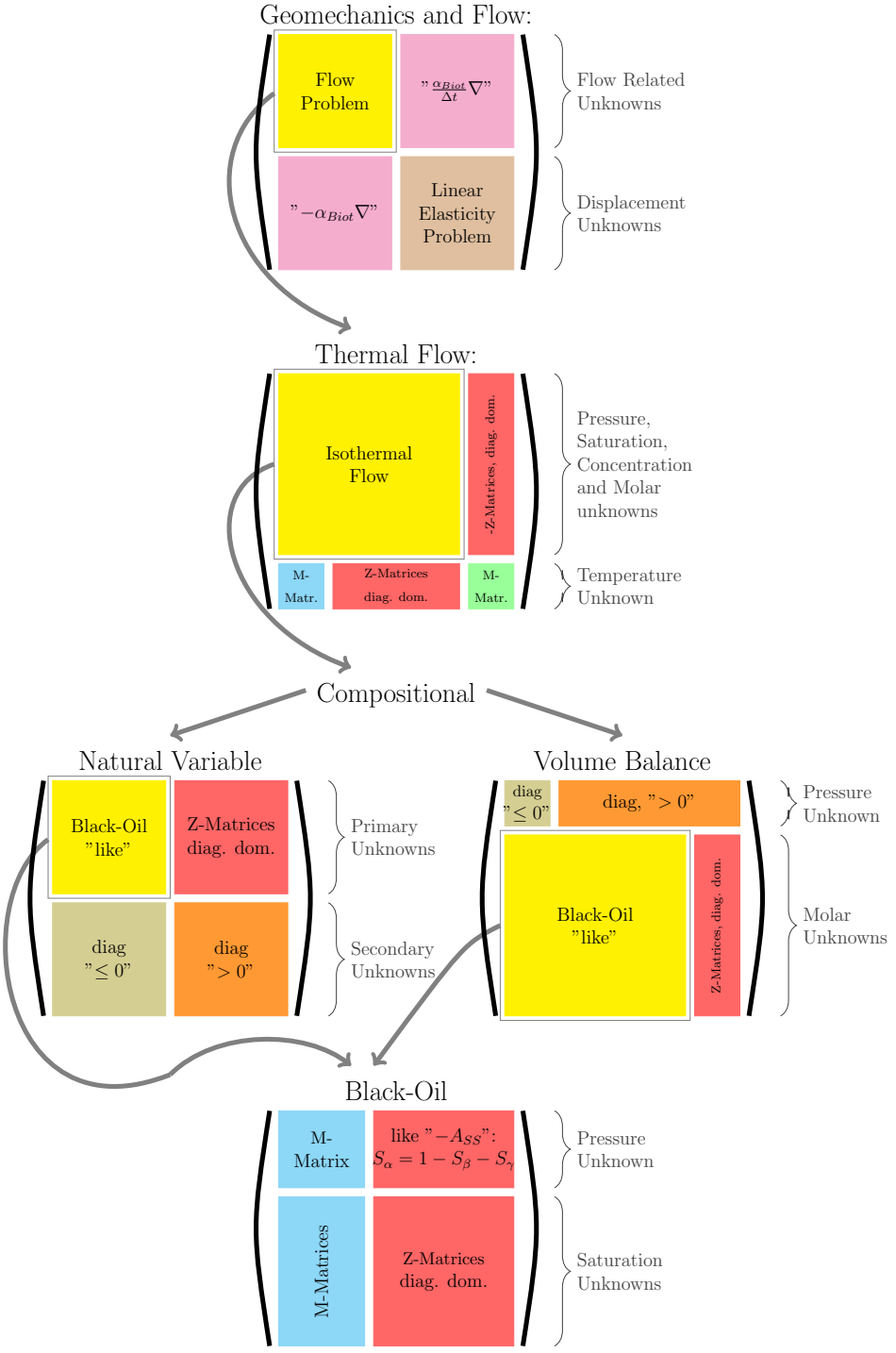


FIGURE 2.10: Schematic visualization of the matrix properties of linear systems resulting from reservoir simulations. The flow chart illustrates how the linear systems from simulations that take more effects into account evolve from the systems from simulations that involve less effects (the resp. yellow blocks give the links). Matrix blocks of the same background (and therefore with comparable properties) share the same color.
All claimed properties hold under certain conditions on time step size and source terms.

Chapter 3

System-AMG for Model FIM Black-Oil Simulations

In the previous chapter we have seen that reservoir simulation models of different complexity lead to linear systems with different properties. However, the Black-Oil model can be seen as a base case. The linear systems resulting from more complex simulations always contain a sub-part that has the same properties as the Black-Oil systems. Therefore, we will first consider an AMG-approach for this base case, before we extent it to more complex models.

As we have discussed in Section 2.3, Black-Oil models describe three-phase flow within porous media. They still are heavily used in the oil industry and there is a large number of individual linear systems to be solved during a simulation, one for each Newton iteration of each time step. Hence, the solution of linear systems typically is by far the most time consuming part of a simulation. It can cover more than 80% of the run time of today's industrial reservoir simulators [46].
An efficient solver does not only need to be fast, it also needs to be scalable with the problem size. As of today, the giant reservoirs in the middle east are discretized in grids with a billion cells and more. As this still results in resolutions of dozens of meters, a further increase can be expected as soon as the computer hardware can handle the amount of data.
Because the simulation of flow in a porous medium corresponds to a diffusion dominated problem (cf. the flux part of the mass balance (2.10)), multigrid methods are natural candidates to consider when designing efficient methods. However, a solver method has to take the different physical unknowns into account, and the

corresponding matrix properties that they induce in the Jacobian matrices (2.14). Also for a given modeling approach, the high number of physical effects may have significant influences on the matrix properties, as described in the previous chapter. In practice, especially source terms may cause difficulties for the linear solver, if they are not treated properly. We will discuss this aspect, as well as the general question of how to ensure robustness of the solver approach in complex industrial applications, in Chapter 4.

In this chapter we assume somewhat mild conditions for the Black-Oil model that the linear systems we consider result from. More precisely, we assume the conditions of Theorems 2.21, 2.25 and 2.28 to hold. That is, the pressure sub-problem A_{pp}, as well as the blocks of A_{sp}, are M-matrices. A_{ss} and A_{ps} are (block) diagonally dominant.

This way, the solver approach becomes transparent and also allows for some convergence considerations. However, prior to discussing the System-AMG approach for Black-Oil problems, we briefly review the System-AMG methodology.

3.1 Review of Algebraic Multigrid Methods

Multigrid methods exploit a hierarchy of grids to achieve numerical efficiency for the solution of linear problems that have certain properties. They are especially suited for problems that result from the discretization of diffusion processes. In *geometric multigrid* (GMG) approaches, the hierarchy is induced by the geometry, e.g., discretizations at smaller grid resolutions. In contrast to this, *algebraic multigrid* (AMG) approaches construct the hierarchy as part of the algorithm, based on certain matrix properties. This makes them being independent of geometric information and allows to deal with systems that result from unstructured grids and contain strong heterogeneities. These aspects are of particular importance in reservoir simulation.

We therefore do not consider geometric multigrid approaches, as for the problems that we are concerned with the generation of a geometrical grid hierarchy is highly difficult, if not practically impossible.

3.1.1 Scalar-AMG

Before discussing AMG approaches for linear problems that involve different physical unknowns, i.e. coupled systems of linear problems, we review the idea of AMG for scalar problems. The application to systems then is an extension of the underlying ideas. In this thesis we will only give a brief survey to an extent needed to motivate the application of AMG in reservoir simulations and to discuss possible limitations and difficulties. For a detailed description of AMG and the theory behind it, we refer to the literature [15, 85, 95].

We start our description with the general idea of multigrid with geometric interpretations for exemplification. We will then describe how AMG methods construct the different operations independently of the geometry.
Let us consider the linear problem

$$Ax = f, \tag{3.1}$$

where we assume A to be a sparse, symmetric and positive definite M-matrix. This is the prototype linear system to be solved by Scalar-AMG and there exists a rigorous convergence theory for the respective two-level method. We note that this theory can be extended to matrices of essentially positive type (cf. Definition 2.32 in Section 2.3.2.2). While mere positive definiteness already is sufficient to achieve convergence, this does not guarantee a convergence speed independent of the dimension of A. We refer to the above mentioned literature for details. Linear systems with the described properties result, for instance, from classical discretizations of elliptic PDEs. Let us assume that our problem results from such a discretization on a grid of grid size h. We therefore also write A_h instead of A. The outcome of an AMG iteration is an error approximation from the defect equation

$$A_h e_h^i = r_h^i =: f_h - A_h x_h^i, \tag{3.2}$$

where x_h^i is the i-th solution iterate and r_h^i, e_h^i are the corresponding residual and error vector. In the first iteration we start with an initial guess x_h^0.
In geometric multigrid, we interpret the error to be expanded in a Fourier series. In this sense, it is composed of different frequencies. The efficiency of multigrid methods for our considered class of problems results from tackling these components differently, depending on whether they are considered high- or low-frequency.

More precisely, with a smoothing step we aim at reducing high-frequency components, which is combined with a coarse grid correction process to also reduce low-frequency components efficiently. In principal, this proceeding is the same in algebraic multigrid.

3.1.1.1 Smoothing Process

The purpose of an iterative smoothing process in multigrid cycles is to reduce high-frequency error components. We denote the *smoothing operator* as

$$S_h = \mathbb{1} - Q_h^{-1} A_h, \tag{3.3}$$

where the subscript identifies it with one of the levels of the hierarchy. Q_h is specified depending on the utilized method. Classically, this is a relaxation method like ω-Jacobi or Gauss-Seidel, where Q_h is the (weighted) diagonal of A_h and the lower triangular part of A_h, including the diagonal, respectively. Also other one-level methods may be used as a smoother. For incomplete factorization (ILU) methods this is, for instance, described by Wittum and Oertel [79, 108]. We will get back to this in Section 6.1.1 and see that for our type of problems ILU smoothers allow for faster AMG approaches.
All smoothing methods have in common that, while computationally they are relatively cheap per iteration, they converge (very) slowly for problems like (3.1) - if A_h was not diagonally dominant. This slow convergence speed is caused by low-frequency error components being reduced slowly. High-frequency components are reduced very efficiently within only a few iterations, independently of the problem size. In the geometric multigrid context, an error with drastically reduced high-frequency components is a rather smooth function. In the AMG methodology this interpretation does not need to hold. Here an error is considered smooth if it cannot be efficiently reduced further by the smoothing method.

3.1.1.2 Coarse Grid Correction

While the smoother reduces high-frequency error components efficiently, this is typically not the case for low-frequency error modes. Low-frequency modes, however, can be appropriately captured on a grid of a lower resolution $H > h$ as well. At this lower resolution they might become high-frequency modes and can

be reduced efficiently by the smoother. In the context of AMG we do not have a coarser grid and a corresponding operator, but we automatically construct *coarse level operators*, denoted by A_H (cf. (3.4) below).
The idea of a two-level iteration step is to transfer the residual vector from the initial h-level to the coarser H-level by a *restriction operator* I_h^H. We then solve for the error in the coarse level problem and transfer it back to the initial level. This transfer is achieved by an *interpolation operator* I_H^h. The concrete interpolation weights depend on the matrix entries in A_h, which allows AMG to adapt the interpolation to a particular matrix.
We typically choose the restriction to be the transpose of the interpolation operator, i.e. $I_h^H = (I_H^h)^T$, and use the *Galerkin operator* as coarse level operator:

$$A_H = I_h^H A_h I_H^h. \tag{3.4}$$

Obviously, A_H is spd if A_h is. We can now express the resulting coarse grid two-level correction operator as:

$$K_{h,H} = \mathbb{1} - I_H^h A_H^{-1} I_h^H A_h. \tag{3.5}$$

The use of the Galerkin operator induces a variational principle [95]: The coarse grid correction minimizes the energy norm[1] of the error w.r.t. the interpolation of all variations of the error e_H. Clearly, the minimum depends on the interpolation operator I_H^h and the quality of the interpolation is crucial for AMG's convergence speed.
As a further consequence of the variational principle, in the energy norm AMG is guaranteed to converge if only the smoother converges.

3.1.1.3 From Two-Level to Multilevel

The smoothing and coarse grid correction operators, S_h and $K_{h,H}$, target at reducing different error components efficiently. While the smoothing operator reduces high-frequency components, the low-frequency ones are reduced by the coarse grid correction operator. By using ν_{pre} and ν_{post} smoothing steps before and after the coarse grid correction, respectively, the operator that defines a two-level iteration reads as:

$$M_{h,H} = (S_h)^{\nu_{post}} K_{h,H} (S_h)^{\nu_{pre}}. \tag{3.6}$$

[1] The energy norm is induced by the scalar product $< A_h ., . >_{std}$.

The coarse grid problem A_H, although of lower dimension than A_h, may still be too big to be inverted efficiently. Further two-level processes can recursively be used to approximate the solution of the coarse grid problem. This proceeds till the dimension of the coarsest problem, A_C, is small enough to use a direct solver efficiently. This process is sketched in Figure 3.1 for a so-called V-cycle. In this thesis, we will not consider different types of cycles that employ more than one coarse grid correction per level (e.g., W- or F-cycles).

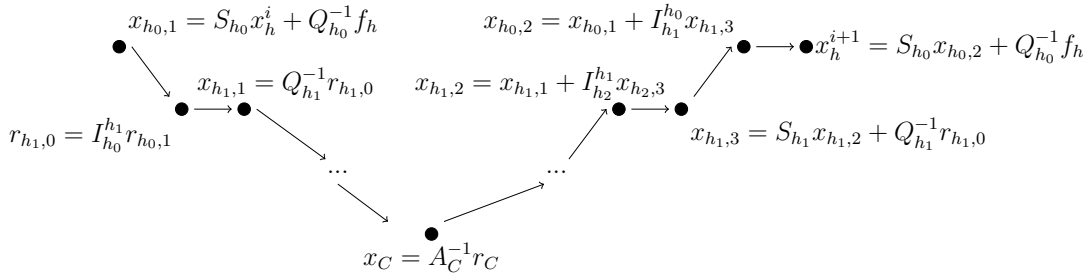


FIGURE 3.1: Schematic illustration of a V-cycle for solving $A_h x_h = f_h$, with one pre- and post-smoothing sweep per level. We assume pre- and post-smoothing method to be used.

3.1.1.4 Construction of the Hierarchy

Algebraic multigrid approaches shall not rely on geometric information. Hence, it is not possible to use coarse grid analogues of the finest grid to define coarser levels. Instead, the hierarchy is constructed based on the matrix A_h itself in a setup phase. Hereby, the algebraic multigrid principle is independent of whether the linear system (3.1) results from the discretization of a PDE. As long as A_h has the properties AMG seeks to exploit, it may also result from a different background. The concrete coarsening depends on the coupling structure of the matrix A_h. This is reflected by the adjacency graph, G_A (cf. Definition 2.10).
The *strength of the couplings* will play an important role. For a particular off-diagonal matrix entry a_{ij}, it is defined as [85, 95]:

$$str(i,j) := \frac{-a_{ij}}{\max\limits_{k \neq i}(-a_{ik})} \text{ with } i \neq j. \tag{3.7}$$

We define an off-diagonal matrix entry a_{ij} to correspond to a strong connection, if the respective measure of strength is above a certain threshold, ϵ_{str}. This allows

to define the *strong adjacency graph* G_{str} from G_A as

$$G_{str} = (V_A^G, E_{str}^G) \text{ with } E_{str}^G = \{e_{ij}^G \in E_A^G | i \neq j, str(i,j) \geq \epsilon_{str}\}. \tag{3.8}$$

We have $0 \leq \epsilon_{str} \leq 1$ and typically 0.25 is chosen.

In classical AMG, hierarchies are constructed based on a C/F-splitting. That is, we label each vertex in the strong adjacency graph with C or F, where C denotes those vertices that correspond to values that are represented on the coarser level. In the coarse grid correction process of AMG, the correction at C-values is directly obtained from the coarse level. The correction at F-values is obtained by interpolating from strongly connected C-values. It is crucial that, after the smoothing by S_h, the error varies smoothly between all F-values and the respective C-values. It can be shown that this just holds for strong connections. Hence, the C-values are selected as a subset of the vertices in the strong adjacency graph G_{str}. With appropriate interpolation formulas, i can be interpolated from j, if i was an F and j a C-value.

We have not yet mentioned how the C-values are selected in G_{str}. We will only consider the classical *Ruge-Stüben (RS) coarsening* in this thesis. In Section 6.1.1 we will also discuss *aggressive* variants of this approach that will allow for faster multigrid cycles and less memory requirements.
The classical coarsening algorithm selects C-values such that they form a maximal independent set of maximal size in G_{str}. Hence, each F-value is strongly coupled to several C-values. This is of particular importance in reservoir simulations, as we are concerned with highly heterogeneous matrices. The more C-values are incorporated in the interpolation of an F-value, the better the complex coupling structure of the matrix can be taken into account.

Remark 3.1. *We note that the coarsening process also allows for so-called forced-F-values. They are not required to be strongly coupled to any C-value, as they are not included in the coarse grid correction process. Forced-F-values may either be user supplied (e.g., well equations; cf. Remark 2.34), or selected by the algorithm itself. For instance, for rows where the diagonal strongly dominates the off-diagonals, the smoother does not only smooth, but sufficiently solve for the respective variable. No coarse grid correction is needed then and these variables are natural candidates for forced-F-values.* △

Remark 3.2. *There is an alternative coarsening approach, the* aggregation *AMG. Here, the coarse level is defined by grouping the vertices in the strong adjacency graph into aggregates. This does not allow for any unknown at the finest level to be interpolated from several ones at the coarse level. Moreover, additional efforts, e.g., smoothing the interpolation, K- and/or V*-cycles, are necessary in order to give h-independent convergence. We refer to the literature [77, 99] for further discussion.*
Except for a special comparison, we do not further consider these approaches in this thesis. This is because for the highly heterogeneous problems from reservoir simulations, the interpolation is less robust than the one from the RS-coarsening. △

3.1.1.5 AMG in Practical Applications

In practice, AMG is typically not applied as a stand-alone solver, but as a preconditioner for a Krylov-method (see Saad [88] for a discussion on such methods). Due to its robustness and its applicability to non-symmetric problems, like the full problems from reservoir simulations, in this thesis we will use flexible GMRes.

For our model problem (3.1) A_h is a symmetric, positive definite, sparse M-matrix. In Theorem 2.21 we have seen that this exactly holds for A_{pp} in incompressible simulations with TPFA discretizations and appropriately located wells.
In Section 2.3.2.2 we have seen that with MPFA discretization schemes A_{pp}, due to small positive off-diagonal entries, might not be an M-matrix anymore. We have discussed that we can expect it to be of essentially positive type (cf. Definition 2.32) then. We note that the AMG theory can be extended to this class of matrices. In the compressible case, under certain assumptions on the time step size and the wells' source terms, Theorem 2.25 showed that A_{pp} still is an M-matrix, but not necessarily symmetric anymore. However, the symmetric part of A_{pp} under these assumptions still is positive definite. Moreover, we can expect the matrices to be only "slightly" non-symmetric. The lack of symmetry only results from the upstream-weighting of the transmissibilities, which implies a non-symmetric dependence on the pressure field. The transmissibilities themselves, which can be expected to dominate the description of the fluid flux, are symmetric. We refer to the discussion in Section 2.3.2.1. According to practical experience, this minor lack

of symmetry does not cause problems for AMG's applicability, though a rigorous theory is not available for this case.

3.1.2 System-AMG

The situation is more involved if different physical unknowns are considered within one *coupled linear system*

$$A_{system}x = \begin{pmatrix} A_{11} & \cdots & A_{1k} \\ \vdots & \ddots & \vdots \\ A_{k1} & \cdots & A_{kk} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix}. \tag{3.9}$$

Here k denotes the number of physical unknowns. The A_{ij} are matrices by themselves and x_i, f_i are vectors.
Consider the Jacobian from Black-Oil simulations (2.14): The full system is far from being a symmetric, positive definite matrix, let alone a symmetric M-matrix. Nevertheless, the fluid flow is governed by a diffusive process, which gives rise to the expectation that it should be possible to extent the basic idea of AMG.
Our approach of extending AMG to coupled systems follows the ideas from Ruge, Stüben and Clees [26, 85]. We will only give a brief review here. There are essentially two such approaches, the *unknown-wise* and the *point-wise*. Which of them is suitable depends on the matrix properties of a particular application.

3.1.2.1 Unknown-Wise Approach

A natural extension of the Scalar-AMG idea is the individual application to each of the different unknowns. This means, the C/F-splitting and the construction of the interpolation operators are done as if Scalar-AMG was applied to the system matrix without cross-couplings between the k different unknowns:

$$A_{Dsystem} := \begin{pmatrix} A_{11} & & 0 \\ & \ddots & \\ 0 & & A_{kk} \end{pmatrix}. \tag{3.10}$$

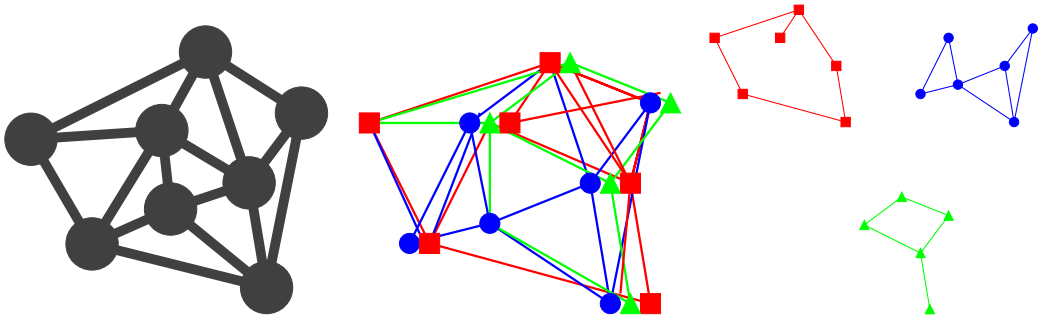
Figure 3.2c illustrates the individual adjacency graphs that correspond to the different unknowns. This is the adjacency graph corresponding to $A_{Dsystem}$.

Given that each A_{ii} satisfies the requirements of Scalar-AMG, the construction of C/F-splitting and interpolation works analogously to what we have described before. However, for the coarse grid problem, we do not compute the Galerkin operator based on $A_{Dsystem}$, but on A_{system}. This way, the couplings between the different unknowns are taken into account on the coarser levels.

For this approach to work properly, it is not only necessary that the diagonal blocks A_{ii} meet Scalar-AMG's requirements, but also that the couplings between unknowns are not "too strong". Only then, individual smoothing per unknown implies overall smoothness of the error and the constructed hierarchy can be reasonably combined with the smoother. More precisely, we require couplings between different unknowns to be relatively small. According to Clees [26], with $\mathcal{A} := A_{system}^{-1} A_{Dsystem}$, we can measure the strength of these couplings as:

$$\rho_u := \rho(\mathcal{A})\rho(\mathcal{A}^{-1}), \tag{3.11}$$

which is greater than, or equal to one.



(A) Point-Wise (B) Full Adjacency Graph (C) Unknown-Wise

FIGURE 3.2: Exemplary illustration of the general System-AMG approaches for a problem with three physical unknowns. (B) shows the full adjacency graph of the problem. The figures (A) and (C) on the left and right show the structures that are relevant for the point-wise and the unknown-wise approach, respectively. The induced point structure, i.e., the adjacency graph of the primary matrix, is shown in (A). (C) shows the adjacency graphs of the three sub-problems.

3.1.2.2 Point-Wise Approach

An alternative approach attempts to formally stay closer to geometric multigrid, for which the same hierarchy would be used for all physical unknowns. For algebraic multigrid this means the same C/F-splitting is used for each of the unknowns. This splitting is constructed based on a so-called *primary matrix* P that describes the matrix connectivity "in terms of points". Given that P meets its requirements, the ideas from Scalar-AMG can be used. With n_p points, P is a sparse matrix of dimension $n_p \times n_p$. The adjacency graph is illustrated in Figure 3.2a for an exemplary problem.
While all physical unknowns of the system are subject to the same C/F-splitting, the interpolation operator can still be constructed individually for each unknown. Whether or not this flexibility needs to be exploited, depends on the particular application.
The primary matrix P has to be chosen such that it represents the strength of the couplings between points sufficiently well. Only then, a C/F-splitting based on P yields a reasonable hierarchy for the entire problem. This requirement can lead to different definitions of P, depending on the concrete application. If, for instance, one of the k unknowns, l, is dominating the overall process, it may be a reasonable choice to use the respective diagonal matrix block, A_{ll}, as primary matrix. However, especially if there is no "outstanding" unknown, constructing P based on norms of the $k \times k$ coupling blocks is another option. For details we refer to the discussion by Clees [26].

3.2 System-AMG Approach for Black-Oil Simulations

In the following we are going to discuss a System-AMG approach to solve linear systems from Black-Oil simulations (cf. ((2.14)) and (2.15)):

$$Jx := \begin{pmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{pmatrix} x = f. \tag{3.12}$$

We have investigated the properties of these systems in Section 2.3, where we have seen that we have to deal with three different physical unknowns. Namely, one

pressure and two saturations. As outlined at the beginning of this chapter, we assume conditions for the model such that A_{pp} is an M-matrix and A_{ss} consists of Z-matrices and is strongly block diagonally dominant. That is, if we denote the sub-blocks of A_{ss} as A_{ss}^{ij}, then all blocks A_{ss}^{ii} are M-matrices that are strongly diagonally dominant in each row. Off-diagonal blocks A_{ss}^{ij} are Z-matrices and whether they are also diagonally dominant depends on the model formulation. If not, we can still expect the diagonal of A_{ss}^{ii} to be strong enough to also dominate the A_{ss}^{ij} blocks.
The properties of A_{sp} and A_{ps} correspond to the ones of A_{pp} and A_{ss}, respectively.

3.2.1 Definition of the Approach

The flux is the fluid's reaction on pressure differences in order to achieve an equilibrium. In this sense, the pressure is "driving" the fluid flux. It is therefore natural to construct an AMG hierarchy based on the pressure sub-problem, which is described by an M-matrix under the assumed conditions. Hence, if we use it as a primary matrix in the context of a point-wise System-AMG approach, we can directly compute a C/F-splitting.
Using the pressure-based C/F-splitting and interpolation also for the saturations mixes different physical unknowns. According to practical experience, this results in a change of the properties of the coarse level problems compared to the initial one: As each pressure variable that remains at the finest level is interpolated from several coarse level variables, each column of the interpolation operator involves multiple non-zero entries. If we constructed the Galerkin coarse level operator for the saturation related sub-blocks with this pressure related interpolation, the diagonal dominance of A_{ss} and A_{ps} typically gets (more and more) lost within the hierarchy. That is, the properties of the full coarse level problems may drastically differ from the ones of the initial problem.
However, due to the outlined diagonal dominance of A_{ss}, there is no need to apply a hierarchical treatment to the saturations. Hence, given that the correct pressure was known, any iterative solver was able to efficiently solve for the saturations. This especially holds for the fine level smoother.
Thus, we construct the AMG hierarchy based on the pressure sub-problem from the Jacobian (3.12) and leave the saturation unknowns at the finest level. As only the pressure unknown is considered for the hierarchy, from level two and beyond there is no difference to Scalar-AMG. Let I_p be the interpolation operator that is

constructed based on the pressure sub-problem, i.e., the primary matrix. As described before, we choose the restriction to be the transpose of the interpolation. The interpolation operator from the second to the first level, I_H^h, and the Galerkin coarse level operator for the second level, A_H, read as:

$$I_H^h = \begin{pmatrix} I_p \\ 0 \end{pmatrix} \text{ and } A_H = (I_H^h)^T J I_H^h = (I_p)^T A_{pp} I_p. \tag{3.13}$$

If A_{pp} is positive definite, this holds for A_H as well[2]. In the incompressible case, from Theorem 2.21 we know that A_{pp} is symmetric positive definite. In the compressible case, according to Theorem 2.25 it may be a non-symmetric M-matrix. However, as we assume the theorem's requirements to be fulfilled, the symmetric part A_{pp}^{sym} is a symmetric M-matrix, which is positive definite. Therefore, also A_{pp} fulfills the definiteness condition[3].
Hence, it is reasonable to use a standard AMG approach for the second level problem. Here, however, we limit our discussion to the two-level case. The coarse grid correction operator reads as:

$$K_{h,H} = \mathbb{1} - I_H^h A_H^{-1} (I_H^h)^T J. \tag{3.14}$$

We postpone a discussion on the smoother at the finest level to the end of this section.

Remark 3.3. *Clees and Ganzer [27] also suggested to construct a point-based AMG approach with the pressure sub-problem as primary matrix. They applied the hierarchy to pressure and saturations and needed to use ILUT as a smoother. The necessity for a strong smoother like ILUT may have resulted from the loss of diagonal dominance for the saturation related parts that we have outlined above.* △

We find the same System-AMG approach, if we follow the unknown-wise approach. An individual coarsening for the diagonal sub-blocks[4] A_{pp} and A_{ss} and respective interpolation operators would need to be constructed. For the pressure this leads

[2] $\forall v \neq 0 :< A_H v, v >=< A_{pp} I_p v, I_p v >> 0$.

[3] Due to the symmetry of the Euclidean scalar product, we have $\forall v \neq 0 :< A_{pp} v, v >= \frac{1}{2}(< A_{pp} v, v > + < A_{pp} v, v >) = \frac{1}{2}(< A_{pp} v, v > + < A_{pp}^T v, v >) =< A_{pp}^{sym} v, v >> 0$.

[4] Strictly speaking, we would need to consider the diagonal sub-blocks of A_{ss}. Because the Black-Oil model involves three physical unknowns, A_{ss} still is a system for two of them. However, while the argumentation would involve more subscripts but leads to the same result, we consider 'the saturations' as one unknown here.

to the same process as above. For the saturation our C/F-splitting directly decides not to construct any hierarchy. Due to the strong diagonal dominance, it considers the smoother to be sufficient to solve for the saturations. Hence, all saturations are labeled as forced-F-values by the C/F-splitting (cf. Remark 3.1).
As outlined earlier, the unknown-wise approach for a symmetric positive definite system matrix should work well as long as the cross-couplings between unknowns are not too strong. According to Clees [26], (3.11) gives an indication of this strength. This measure itself does not require symmetry and for our system (3.12) it reads as:

$$\rho_u = \rho(\begin{pmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{pmatrix} \begin{pmatrix} A_{pp} & \\ & A_{ss} \end{pmatrix}^{-1})\rho(\begin{pmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{pmatrix}^{-1} \begin{pmatrix} A_{pp} & \\ & A_{ss} \end{pmatrix}). \tag{3.15}$$

Let us consider a simplified model problem with only two phases, where we assume the material properties (density, viscosity, etc.) to be just scaled by constants, and to have the same dependency of them on pressure and saturation. For such a problem, the pressure and saturation derivatives for both mass balance equations are not just somehow related, but only scaled by constants. With $\xi, \eta > 0$ the Jacobian simplifies to

$$J^{simple} = \begin{pmatrix} A_{pp} & -\eta A_{ss} \\ \xi A_{pp} & A_{ss} \end{pmatrix} \quad J_D^{simple} = \begin{pmatrix} A_{pp} & \\ & A_{ss} \end{pmatrix}. \tag{3.16}$$

Here J_D^{simple} is the block diagonal matrix the unknown-wise AMG coarsening is based on (cf. Equation (3.10) in Section 3.1.2.1). The negative sign in the upper right block of J^{simple} results from the fact that the first saturation was expressed by the second one as $S_1 = 1 - S_2$ and then the derivative with respect to S_2 was computed (cf. Remark 2.31). We can now show that the coupling between both unknowns is minimal in the sense of (3.11). The respective proof will make use of the following lemma.

Lemma 3.4. *(Theorem 3 from Silvester [93])*
Let a block-matrix $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ *with* $A, B, C, D \in \mathbb{R}^{m \times m}$. *If* $CD = DC$, *then for the determinant there is* $det(M) = det(AD - BC)$. △

Lemma 3.5. *With the simplified model problem* J^{simple}, *the strength of the couplings between the unknowns (3.11) is* $\rho_u = 1$.

Proof. Let $\mathcal{J} := J^{simple}(J_D^{simple})^{-1}$ and $\lambda_{min}, \lambda_{max}$ $\mathcal{J}$'s eigenvalues of minimal and maximal absolute value. Then the measure of strength for $\mathcal{J}$ is:

$$\rho_u = \rho(\mathcal{J})\rho(\mathcal{J}^{-1}) = \frac{|\lambda_{max}|}{|\lambda_{min}|}. \tag{3.17}$$

With $\lambda \in \mathbb{R}$ and the above lemma we find:

$$det(\mathcal{J} - \begin{pmatrix} \lambda\mathbb{1} & \\ & \lambda\mathbb{1} \end{pmatrix}) = \det(\begin{pmatrix} \mathbb{1} - \lambda\mathbb{1} & -\eta\mathbb{1} \\ \xi\mathbb{1} & \mathbb{1} - \lambda\mathbb{1} \end{pmatrix}) = det((\mathbb{1} - \lambda\mathbb{1})^2 + \xi\eta\mathbb{1}).$$

Hence, we have $|\lambda_{max}| = |\lambda_{min}|$, which, according to (3.17), implies $\rho_u = 1$.

□

With a general Deal-Oil model we have to expect $\rho_u > 1$. However, the only difference to the simplified problem is that the material properties of the different phases differ. The basic matrix properties still are the same for the pressure and the saturation related blocks of the Jacobian, as we have seen in Section 2.3.2.1. Hence, we expect the coupling between the different unknowns still to be acceptable in terms of our measure ρ_u. As the Black-Oil model just involves an additional saturation unknown, we expect this to hold there as well.

We have not yet discussed the smoother for our System-AMG approach. At coarser levels of the hierarchy, where we only have to deal with pressure unknowns, there is no difference to smoothing in the Scalar-AMG case. This is different at the finest level, where also the saturation unknowns need to be considered. In simple cases, relaxation methods may provide a sufficient smoothing also here. This especially holds if A_{pp} and the blocks in A_{sp} are "very" comparable, i.e., the pressure related properties of all considered phases are nearly the same. However, in practical applications the conditions may be less ideal and the matrix properties may be influenced from various sources, as we have seen in the previous chapter. The pressure may then have to be adjusted to (slightly) different A_{pp} and A_{sp}. Relaxation methods like Gauss-Seidel may fail in providing a good smoothing and even divergence of the full cycle is observed. Therefore, we switch to ILU-smoothing, where such problems are not observed. In reservoir simulations this seems particularly natural, as methods like ILU(0) or block-ILU(0) are generally used as a default solver method when no AMG is involved. Under mild assumptions, Kwok [61] could prove that block-ILU(0) does converge for Dead-Oil problems, i.e. two-phase problems that essentially have the same properties as Black-Oil problems

have. In Remark 7.7, we will see that we can express block-ILU(0) as an ILU variant with a particular pattern under certain conditions. This typically holds in reservoir simulations. Therefore, we denote S_{ILU} the ILU operator for the problem described by J, with a given non-zero pattern.
In the scope of this thesis we consider standard ILU(0), which turns out to be sufficient. However, we note that the numerical properties of incomplete factorizations can be improved by specific reorderings. This is not subject of study here and we refer, for instance, to Kwok [61] for details.

As ILU(0) is a rather strong smoother, we could use S_{ILU} also as a pre-smoother in order to further improve the pressure correction. However, because our multigrid cycle does not involve the saturation unknowns in the hierarchy, there is no need for a pre-smoothing for them. According to practical experience, the improvement of the convergence rate by using S_{ILU} as a pre-smoother is usually much too small to outweigh the additional computational efforts. Hence, we apply a pre-smoothing to the pressure only. As this does not involve any cross-couplings between unknowns and the pressure sub-problem is an M-matrix, we can expect classical relaxation methods to provide a sufficient smoothing. Let us denote S_{pp} a relaxation operator for the problem that is described by A_{pp}. Then, with the coarse grid correction (3.14), the full System-AMG operator reads as:

$$M_{System-AMG} = M_{h,H} = S_{ILU} K_{h,H} \begin{pmatrix} S_{pp} & 0 \\ 0 & \mathbb{1} \end{pmatrix}. \tag{3.18}$$

The described System-AMG approach is realized in Fraunhofer's SAMG library [96]. In Section 7.2 we will discuss some aspects of the practical realization that are of a general concern for an efficient realization of the described approach in a general multigrid framework. The approach is summarized in Figure 3.3. We will use this System-AMG approach as a basis throughout this thesis. Apart from the described settings, the defaults of the SAMG library, as of version 27z1, are used.

3.2.2 Relation of System-AMG and CPR-AMG

A widely used approach to solve linear systems (3.12) from Black-Oil simulations is the *Constrained Pressure Residual* (CPR) method, proposed by Wallis [101, 102]. This method, as each iteration consists of two successive preconditioning

Setup	**Cycle**
• Saturation unknowns remain at the finest level • RS coarsening, $\epsilon_{str} = 0.25$ • Forced-F for rows w/ diag. dom. by a factor 100 • Direct solver at the coarsest level	• $V_{1,1}$ cycle • C/F Gauss-Seidel smoother for the pressure • Post-smoother at the finest level: ILU(0) • Preconditioned FGMRes, restarted after 30 iterations

FIGURE 3.3: Overview of the System-AMG approach that we are going to use for Black-Oil problems - if nothing else is mentioned. It is realized with Fraunhofer's SAMG library, version 27z1. All further parameters remain at their defaults.

operations, is also referred to as two-stage method. Let us denote the respective iteration operators as M_1 and M_2. Then the CPR iteration operator reads as $M_{CPR} = M_2 M_1$.
The two stages are chosen due to the different properties of the linear sub-systems:

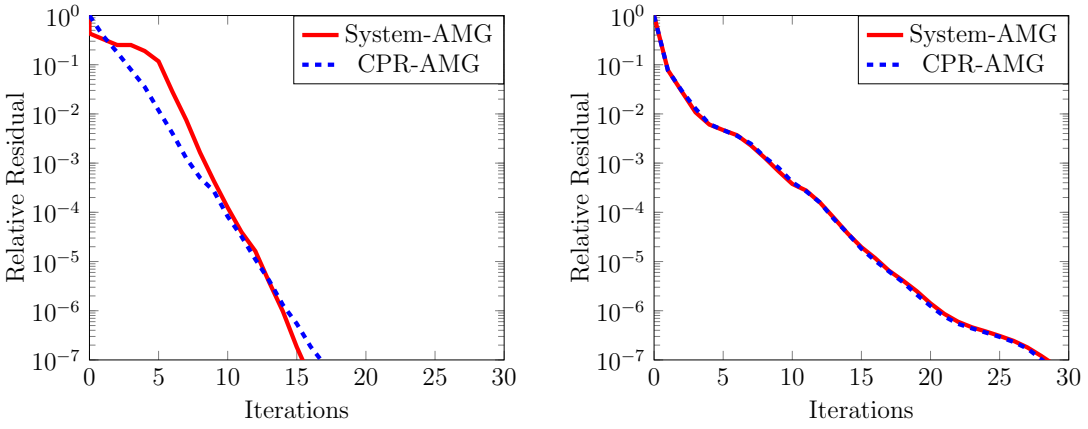
1. In the first stage, based on the scalar pressure sub-problem that is described by A_{pp}, an approximate pressure solution is computed. That is, the current solution iterate is restricted to the pressure related entries, the pressure problem is approximately solved and the obtained approximation is prolongated back to the full problem. Investing this particular effort regarding the pressure problem is motivated from the fluid flow being driven by pressure differences. Hence, updating the pressure is considered to be the most important task for the solver. In principle, any solver method could be used. Due to its known efficiency for diffusion-driven problems, AMG is a common choice. A single cycle typically is used and we denote the method as CPR-AMG. Let us denote the AMG approximation of A_{pp}^{-1} as $M(A_{pp})$. Then the first CPR operator reads as:

$$M_1 = \mathbb{1} - \begin{pmatrix} \mathbb{1} \\ 0 \end{pmatrix} M(A_{pp}) \begin{pmatrix} \mathbb{1} & 0 \end{pmatrix} J. \tag{3.19}$$

2. After the pressure approximation is computed, the saturation unknowns need to be updated. This is typically achieved by an incomplete factorization method like ILU(0). That is, $M_2 = S_{ILU}$ from (3.18).

Remark 3.6. *Typically ([18, 62]), the CPR method is not applied directly to the Jacobian J (3.12), but to a transformed linear system $J_{dcpl} = C_L J$. The scaling*

with C_L aims at decoupling, or at least weakening, the coupling between pressure and saturation, i.e. the sub-matrix A_{ps}. We will discuss such transformations in detail in Section 4.2 and do not further consider them for the moment. △



(A) mx1041 (B) SPE10

FIGURE 3.4: Comparison of the convergence history of System-AMG and CPR-AMG for two representative linear systems from Black-Oil simulations (cf. Appendix A).

It is easy to see that our System-AMG approach results in essentially the same solver approach as the CPR-AMG method does:

- The second stage of CPR exactly corresponds to the fine level smoother of System-AMG: ILU(0) for the full system.
- Regarding the first stage of CPR, let us consider two-level Scalar-AMG for the pressure problem with one pre- and post-smoothing step defined by S_{pp} from (3.18), respectively. By using $A_h = A_{pp}$, the operator is given by (3.6). The interpolation for the pressure clearly is the same as in System-AMG, i.e. I_p. We can now insert the two-grid operator for the pressure in CPR's first stage (3.19), i.e. replace $M(A_{pp})$, and have:

$$M_1 = \mathbb{1} - \begin{pmatrix} \mathbb{1} \\ 0 \end{pmatrix} S_{pp} I_p A_H^{-1} I_p^T S_{pp} \begin{pmatrix} \mathbb{1} & 0 \end{pmatrix} J = \begin{pmatrix} S_{pp} & 0 \\ & \mathbb{1} \end{pmatrix} K_{h,H} \begin{pmatrix} S_{pp} & 0 \\ & \mathbb{1} \end{pmatrix}, \tag{3.20}$$

  where, according to the Galerkin principle, $A_H = (I_p)^T A_{pp} I_p$ and $K_{h,H}$ is the coarse grid correction operator (3.14) from System-AMG.

From these considerations it is clear that, compared to System-AMG, CPR-AMG just involves an additional post-smoothing sweep. Regarding a practical realization, also the explicit extraction of the pressure sub-problem is not necessary in

System-AMG. This, however, does not have an influence on the algorithm.
Consequently, we can interpret CPR-AMG as a System-AMG approach. Figure 3.4 compares both approaches for Black-Oil problems where AMG is directly applicable to A_{pp}. We will discuss more general problems in Chapter 4.
However, although the methods coincide, the point-of-view is different. In this thesis, we choose the System-AMG point-of-view. This allows for the exploitation of all aspects of System-AMG, and AMG in general, in the further discussion on linear solvers in reservoir simulation. This has two advantages:

- In principal, System-AMG is extendable to additional physical unknowns in a rather natural way. In reservoir simulations this is of interest, as the linear systems from more sophisticated simulation models can be seen as extensions of the systems from Black-Oil simulations. This gives rise to the expectation that we will not have to discuss entirely new approaches later on, but just have to incorporate the new unknowns. We will discuss the extension of our Black-Oil System-AMG approach to compositional, thermal and geomechanical simulations in Chapter 5. We will see that the ability of System-AMG to exploit hierarchies for different unknowns at the same time is beneficial.
- Rather than considering only a scalar pressure problem, System-AMG can access all physical information from the full matrix. This is mandatory to ensure a robust applicability, as we will discuss in Chapter 4. However, it also allows to exploit the algorithmic versatility of AMG. Any optimization of System-AMG can directly reflect impacts on the full linear solution process. In Section 6.1.1, we will see that choosing faster coarsening methods is possible for typical reservoir simulation problems without negative impacts on the convergence behavior of System-AMG.

3.3 Convergence Considerations for a Model

After having described the System-AMG approach for Black-Oil problems, we are now going to discuss the convergence of this approach. We will do so for a fully implicit Dead-Oil simulation. The Dead-Oil model is an analogue of the Black-Oil model from Section 2.3, where we distinguish between only two phases, hydrocarbons and water. We can express the Jacobian, which describes the full

linear system to be solved, as a 2×2 block matrix:

$$J_{model} = \begin{pmatrix} A & -B \\ \hat{A} & \hat{B} \end{pmatrix}. \tag{3.21}$$

Following the notation (2.15), there is $A_{pp} = A$, $A_{ps} = -B$, $A_{sp} = \hat{A}$ and $A_{ss} = \hat{B}$. We have seen that, due to a comparable background, the general properties of A_{pp} and A_{sp} are the same, which we express by denoting them as A and $\hat{A}$ in (3.21). The same holds with the saturation related blocks, i.e., B and $\hat{B}$.
Our discussion is independent of whether the model is applied in one, two or three dimensions. However, we only consider an incompressible model and a TPFA discretization. Moreover, following Theorem 2.21, we assume wells to be given so that A is non-singular. From our discussion in Section 2.3, we then know that

- $A > 0$ is a sparse M-matrix (cf. Theorem 2.21).
- $B > 0$ is a sparse, strongly diag. dominant M-matrix (cf. Remark 2.31).
- $\hat{A} > 0$ is a sparse M-Matrix (cf. Corollary 2.22).
- $\hat{B} > 0$ is a sparse, strongly diag. dominant M-matrix (cf. Theorem 2.28).

We are going to investigate the convergence of System-AMG (cf. Section 3.2.1) for the problem (3.21). More precisely, we are going to discuss the convergence properties of the method consisting of the following two steps[5] [6]:

1. The first step, we denote the respective iteration operator by P, solves for the pressure by AMG and does not apply any change to the saturations:

$$P := \mathbb{1} - \begin{pmatrix} Q_{AMG}^{-1} & 0 \\ 0 & 0 \end{pmatrix} J_{model} = \begin{pmatrix} M_{AMG} & Q_{AMG}^{-1} B \\ 0 & \mathbb{1} \end{pmatrix}, \tag{3.22}$$

   where M_{AMG} denotes the AMG iteration operator for the pressure subsystem $Ap = f_p$ and Q_{AMG} is the respective approximation of A. We do not discuss two- or multilevel convergence results regarding M_{AMG} here. Incorporating a grid-transfer for the pressure in our discussion would result in additional complexity in the notation without giving additional results: Since A is an spd M-matrix, the well-known AMG convergence theory [85, 95] can directly be applied regarding M_{AMG}.

[5] This way we, formally, investigate the influence of the ILU(0) post-smoothing separately from the rest of System-AMG. This rest is a complete AMG process for the pressure sub-problem and we can exploit the existing convergence theory.

[6] Note that all convergence considerations directly apply also to CPR-AMG.

2. The second step, we denote the respective iteration operator by S, applies ILU(0) to the full system.

The System-AMG iteration operator is then described by $M = SP$. We are going to discuss the spectral radius, i.e., the convergence properties, of this operator. We will see that these essentially are determined by how efficiently AMG can solve for the pressure and how efficiently ILU(0) can update the saturations. As we can employ well-known results regarding AMG, our discussion is mainly concerned with the operator S.

3.3.1 Introductory Discussion Regarding ILU

With S resulting from an ILU(0) factorization of the full matrix J_{model}, a discussion on the convergence properties becomes difficult. Already in the simplified case of using block-Jacobi with ILU(0) applied separately to the blocks, we have

$$S_{BJ} = \mathbb{1} - \begin{pmatrix} Q_A^{-1} & 0 \\ 0 & Q_{\hat{B}}^{-1} \end{pmatrix} J_{model} = \begin{pmatrix} S_A & Q_A^{-1}B \\ -Q_{\hat{B}}^{-1}\hat{A} & S_{\hat{B}} \end{pmatrix}, \tag{3.23}$$

with Q_A and $Q_{\hat{B}}$ denoting the ILU(0) approximations of A and $\hat{B}$, respectively, and S_A and $S_{\hat{B}}$ denoting the respective ILU(0) operators. Even with this simplified method, useful conclusions regarding the spectral radius of S_{BJ} are hardly possible for our still rather general incompressible Dead-Oil model.

In order to nevertheless gain an insight into ILU's convergence properties, we consider a modified method for the moment, where the factorization consists of two stages:

1. We compute the full LU-decomposition $J_{model} = L^f U^f$ regarding the 2×2 block-system:

$$L^f = \begin{pmatrix} \mathbb{1} & 0 \\ \hat{A}A^{-1} & \mathbb{1} \end{pmatrix} \text{ and } U^f = \begin{pmatrix} A & -B \\ 0 & \hat{B} + \hat{A}A^{-1}B \end{pmatrix}. \tag{3.24}$$

   For the moment we assume $\hat{A}A^{-1}$ to be given and do not discuss how to compute this term.

2. To solve the full system based on (3.24), we need to apply a backwards and forwards solution sweep for the systems that are described by L^f and U^f, respectively. Regarding L^f, as it is triangular, this is trivial. In contrast to this, U^f is only block-triangular. We use ILU(0) to approximately solve the backwards solution sweep.

The above two-stage factorization can be regarded a compromise between a full decomposition and ILU(0). As ILU(0) approximates the full decomposition, we can expect it to also approximate this two-stage factorization.
We note that the full LU-decomposition in the first stage corresponds to decoupling the saturation from the pressure via a Schur complement. That is, we could equally well apply this Schur complement and then solve the resulting system with ILU(0). Hence, we can expect ILU(0) for the full system to approximate this Schur complement.
Due to the block-triangularity of the Schur complement system, i.e. U^f, the convergence properties of the block-Jacobi-like method from (3.23) are determined by those of ILU(0) being applied to the systems that are described by A and $\hat{B} + \hat{A}A^{-1}B$, respectively. The analysis of these two operations is much simpler than for the full system. We will see that the same also holds for a full ILU, rather than block-Jacobi.

However, in the general case of J_{model}, the ILU operation, S, only approximates the two-stage factorization from above. Hence, the convergence properties would only approximate the ones of S and it is not clear to which extent. Moreover, the computation of $\hat{A}A^{-1}$ in an efficient way is a problem for a practical realization. Although A and $\hat{A}$ are somehow related, we generally cannot quantify this relation. We can see a linear relation as the limit case for the relation between A and $\hat{A}$, as well as B and $\hat{B}$. In the next section we are going to see that ILU(0) applied to the original system then is equivalent to ILU(0) for the Schur complement system.

3.3.2 Convergence Properties of ILU in the Limit Case

From our discussion in Section 2.3 we know that A and $\hat{A}$, as well as B and $\hat{B}$ are closely related. In this section we are going to discuss the limit case of a linear relation. More precisely, we assume $\hat{A} = VA$ and $\hat{B} = WB$ with diagonal matrices $V, W > 0$ (non-positive entries would be a contradiction to $A, B, \hat{A}$ and $\hat{B}$ being

M-matrices).
Regarding our model, this assumption only neglects non-linear dependencies between the two phases. However, we expect the linear dependencies to dominate. That is, for instance, the density of oil and water differ by a certain factor per cell, which is mainly determined by the general difference of the oil and water density. In the limit case model, we only neglect small deviations from this general factor, for instance, due to pressure differences.
The linear system matrix then reads as:

$$J_{limit} = \begin{pmatrix} A & -B \\ VA & WB \end{pmatrix}. \tag{3.25}$$

The linear dependence implies $\hat{A}A^{-1} = V$ and we obtain the Schur complement

$$\tilde{J}_{limit} = \underbrace{\begin{pmatrix} \mathbb{1} & 0 \\ -V & \mathbb{1} \end{pmatrix}}_{:=C} J_{limit} = \begin{pmatrix} A & -B \\ 0 & (V+W)B \end{pmatrix}, \tag{3.26}$$

which corresponds to U^f from (3.24).

We note that ILU corresponds to a row-wise elimination of the respective matrix that, as it is incomplete, generally leads to a rest. As each row of $\hat{A} = VA$, however, is a multiple of the respective row in A, the elimination of the lower left block in (3.25) does not lead to a rest, also for the incomplete factorization.
Therefore, the ILU(0) iteration operator for J_{limit} is equal to that of $\tilde{J}_{limit}$. As the latter one turns out to be block-triangular, the computation of the spectral radius simplifies.
We are now going to formally show the equality of the two ILU iteration operators. This exploits the fact that the ILU factorization of a matrix w.r.t. a non-zero pattern is unique under certain conditions, if it exists. Our argumentation here is the same as the one from Meijerink and van der Vorst [71] for M-matrices: they exploited the M-matrix property to prove the existence of the ILU factorization, but not for showing its uniqueness.

Lemma 3.7. *Let $\mathcal{A}$ be an $n \times n$ matrix and assume the incomplete factorization $\mathcal{A} = \mathcal{LU} - \mathcal{R}$ w.r.t. the non-zero pattern $P_{\neq 0}$ exists (i.e., $(ij) \notin P_{\neq 0} \Rightarrow (\mathcal{L})_{ij} = (\mathcal{U})_{ij} = 0$) with a lower triangular $\mathcal{L}$ and an upper triangular $\mathcal{U}$. We assume $diag(\mathcal{L}) = \mathbb{1}$ and $\mathcal{U}$ is regular (i.e., $(\mathcal{U})_{ii} \neq 0 \ \forall 1 \leq i \leq n$).*
Moreover, we require that, with $(ij) \in P_{\neq 0}$, $(\mathcal{LU})_{ij} = (\mathcal{A})_{ij}$, i.e., there is no rest

(from Saad ([88], Proposition 10.2) this is known to hold for a classical ILU(k) factorization).
Then the ILU factorization is unique for this non-zero pattern.

Proof. The $2n^2$ entries of $\mathcal{L}$ and $\mathcal{U}$ are uniquely defined:

- n^2-n entries result from the factors being triangular (i.e., the upper triangle of $\mathcal{L}$ and the lower one of $\mathcal{U}$ are zero).
- n entries result from $\mathcal{L}$ being normalized.
- If we assume $|P_{\neq 0}| = m$, then $n^2 - m$ entries follow from the factors being zero outside the non-zero pattern.
- The remaining m entries follow from the zero rest matrix in the non-zero pattern:
  - As $(\mathcal{L})_{11} = 1$, for $i = 1$ we have $(\mathcal{U})_{1j} = (\mathcal{A})_{1j}$ for any $(1j) \in P_{\neq 0}$.
  - For $i = 2$ we have $(\mathcal{A})_{21} = (\mathcal{L}_{21})(\mathcal{U})_{11}$. As $(\mathcal{U})_{11}$ is already determined, this determines $(\mathcal{L})_{21}$. We analogously continue with $(\mathcal{A})_{22} = (\mathcal{L}_{21})(\mathcal{U})_{12} + (\mathcal{L}_{22})(\mathcal{U})_{22}$, which, due to $(\mathcal{L})_{22} = 1$, determines $(\mathcal{U})_{22}$. This proceeds until $(\mathcal{A})_{2n} = (\mathcal{L}_{21})(\mathcal{U})_{1n} + (\mathcal{L}_{22})(\mathcal{U})_{2n}$ determines $(\mathcal{U})_{2n}$.

  A successive proceeding determines each entry of the factors in the non-zero pattern by the respective entry in $\mathcal{A}$.

□

We can use the above result to describe the ILU(0) factorization of J_{limit}. Let us denote the ILU(0) factorization of A by $L_A U_A$ and the ILU(0) factorization of $(V + W)B$ by $L_{VWB}U_{VWB}$. We assume that both factorizations exist. For A, as it is a symmetric M-matrix, the existence was showed by Meijerink and van der Vorst [71]. $(V + W)B$ does not need to be symmetric but, due to $V, W > 0$ being diagonal and B strongly diagonally dominant M-matrix, $(V + W)B$ is a strongly diagonally dominant M-matrix. According to practical experience, the ILU factorization of this non-symmetric M-matrix exists[7] as well.
We can now state the following lemma regarding the ILU(0) factorization $J_{limit} =$

[7]We could modify the ILU factorization such that its existence is guaranteed: Let $G = (g_{ij}) = (V + W)B$ and $H = (h_{ij})$ with $h_{ii} = g_{ii}$ and $h_{ij} = max(g_{ij}, g_{ji})$. Then $H = G - P$, where P is a non-negative matrix. Due to G being a diagonally dominant M-matrix, so is H (the off-diagonals just became weaker) and, hence ([71]), the ILU factorization $H = Q_H - R_H$ exists and is a regular splitting (i.e., Q_H^{-1} and R_H are non-negative matrices). Consequently, $G = Q_H - R_H - P =: Q_H - R_G$ is a regular splitting as well. For our further discussion it does not matter whether we used this modified factorization regarding the lower right matrix block, or the ILU(0) factorization.

$LU - R$ and determine the different blocks of the ILU factorization that we will use later on.

Lemma 3.8. *For the ILU(0) factorization of J_{limit}, with some appropriate U_2, there is*

$$L = \begin{pmatrix} L_A & 0 \\ VL_A & L_{VWB} \end{pmatrix} \text{ and } U = \begin{pmatrix} U_A & U_2 \\ 0 & U_{VWB} \end{pmatrix}. \tag{3.27}$$

Proof. Let us denote the factors in a general way as:

$$L = \begin{pmatrix} L_1 & 0 \\ L_2 & L_3 \end{pmatrix} \text{ and } U = \begin{pmatrix} U_1 & U_2 \\ 0 & U_3 \end{pmatrix}.$$

By construction it is clear that L_1U_1 is the ILU(0) factorization of A, i.e., $L_1 = L_A$ and $U_1 = U_A$. As the rest matrix is zero in the non-zero pattern, $V > 0$ is diagonal and the factorization is unique, it follows that $L_2 = VL_A$.
In the non-zero pattern, L_1U_2 equals $-B$. As the same holds for $L_2U_2 + L_3U_3 = VL_1U_2 + L_3U_3$ regarding WB, this induces L_3U_3 in the non-zero pattern equals $WB + VB$. Hence, as L_3 and U_3 are triangular and L_3 is normalized, L_3U_3 is the ILU(0) factorization of $WB + VB$, i.e., $L_3 = L_{VWB}$ and $U_3 = U_{VWB}$. □

Let us now compute the ILU(0) factorization $\tilde{L}\tilde{U}$ of $\tilde{J}_{limit}$. Regarding the first blocks, i.e., A and $-B$, there is no difference compared to J_{limit}. Regarding the saturation rows, due to the Schur complement, the derivation of the factors also is straight forward and we have

$$\tilde{L} = \begin{pmatrix} L_A & 0 \\ 0 & L_{VWB} \end{pmatrix} \text{ and } \tilde{U} = \begin{pmatrix} U_A & U_2 \\ 0 & U_{VWB} \end{pmatrix}. \tag{3.28}$$

That is, with C from (3.26) we have $\tilde{L} = CL$ and $\tilde{U} = U$. Both allows to state the following corollary regarding the ILU(0) iteration operators:

Corollary 3.9. *The ILU(0) iteration operators S and $\tilde{S}$ for the linear systems described by J_{limit} and $\tilde{J}_{limit}$, respectively, are equal.*

Proof. $\tilde{S} = \mathbb{1} - \tilde{U}^{-1}\tilde{L}^{-1}\tilde{J} = \mathbb{1} - U^{-1}L^{-1}C^{-1}CJ = S.$ □

Remark 3.10. *In the point-wise numbering, our argumentation would apply per block-row (i.e., the set of rows that correspond to the same point) and we ended with the same result.* △

3.3.3 System-AMG in the Limit Case

We note that regarding the hierarchical treatment of the pressure it does not matter whether we consider the linear system that is described by J_{limit} or $\tilde{J}_{limit}$, i.e. $\tilde{P} = P$. Consequently, also regarding the full solver operation, it does not matter whether we consider the original linear system, or apply the Schur complement before solving the system, i.e. $\tilde{S}\tilde{P} = \tilde{M} = M = SP$.
The above results give

$$\begin{aligned} M = \tilde{M} = \tilde{S}\tilde{P} &= \begin{pmatrix} S_A & X \\ 0 & S_{VWB} \end{pmatrix} \begin{pmatrix} M_{AMG} & Q_{AMG}^{-1}B \\ 0 & \mathbb{1} \end{pmatrix} \\ &= \begin{pmatrix} S_A M_{AMG} & S_A Q_{AMG}^{-1} B + X \\ 0 & S_{VWB} \end{pmatrix}, \end{aligned} \tag{3.29}$$

where S_A and S_{VWB} denote the ILU(0) iteration operator for the systems that are described by A and $(V + W)B$, respectively. X denotes some term that results from $U_2 \neq 0$. This term does not matter for the spectral radius of M.
In our limit case we have now shown that the convergence properties of the full solver approach are determined by how efficiently AMG can solve for the pressure and ILU(0) for the iterations:

Corollary 3.11. $\rho(M) = max(\rho(S_A M_{AMG}), \rho(S_{VWB}))$.

Proof. This is a direct consequence of Lemma 3.4 for determinants of block matrices. □

In the following, we are going to discuss both spectral radii and see that we can expect them to be uniformly bounded away from one within the class of problems at hand. We will find that the overall convergence, i.e. $\rho(M)$, is determined by

- The convergence properties of AMG for the pressure sub-problem in the energy norm.
- The convergence properties of ILU for the saturation sub-problem in the Schur complement system, i.e. $\rho(S_{VWB})$. We will see that these are determined by the diagonal dominance of $(V + W)B$.

We will especially see that AMG is essential for the overall convergence rate being significantly lower than one: Without the operator P, the spectral radius of M was determined by the spectral radii of S_A and S_S. Since A is a weakly diagonally dominant M-matrix, although $\rho(S_A) < 1$, it is close to one. These considerations imply that the applicability of AMG to the pressure sub-problem is crucial. In our model case we have chosen properties that ensure this. However, in a general simulation, the conditions can be less ideal, as we will see in Chapter 4.
After discussing the limit case, we are going to outline the relation of this limit case with the more general case, i.e., the system given by J_{model} and not J_{limit}.

Remark 3.12. *We cannot simply choose $S_A = \mathbb{1}$. Then we generally had $\tilde{L} \neq CL$ and could not exploit $\tilde{S} = S$. For the same reason, we cannot use Gauss-Seidel or Jacobi for J_{model}, but need an incomplete factorization.* △

3.3.3.1 Convergence for the Saturation Part

In this section we are going to discuss the spectral radius of S_{VWB}, which is the iteration matrix that describes the ILU(0) method for the linear system that is described by $(V + W)B$. We have already seen that, due to B being a strongly diagonally dominant M-matrix and $V, W > 0$ are diagonal, $(V+W)B$ is a strongly diagonally dominant M-matrix as well and we can expect any iterative method to efficiently solve the respective linear system. For the Jacobi method, for instance, we have:

Lemma 3.13. *Let $\mathcal{A}$ be an arbitrary matrix that is diagonally dominant in each row. Then for the iteration matrix S_{Jac} that results from the Jacobi method, we have $\rho(S_{Jac}) < 1 - \sigma$, where $\sigma \in (0, 1]$ depends on the strength of the diagonal dominance.*

Proof. As $\mathcal{A}$ is diagonally dominant in each row, we can split it into

$$\mathcal{A} = \mathcal{A}_{\mathcal{M}} + \mathcal{A}_{\mathcal{D}}, \tag{3.30}$$

where $\mathcal{A}_M$ is diagonally dominant in each row and $\mathcal{A}_D$ is diagonal and positive. Let $\mathcal{Z} = diag(\mathcal{A})$ and $\mathcal{Z}_M = diag(\mathcal{A}_M)$, then the positivity of $\mathcal{A}_D$ implies the existence of a $\lambda > 1$ such that $\mathcal{Z} \geq \lambda \mathcal{Z}_M$.
As $\mathcal{A}_M$ is diagonally dominant, the Jacobi method does converge for the linear

system that is described by $\mathcal{A}_M$. The Jacobi method results from the splitting $\mathcal{A}_M = \mathcal{Z}_M - \mathcal{R}$, with a rest matrix $\mathcal{R}$. Hence, we have some consistent matrix norm $||.||_x$, such that $||\mathcal{Z}_M^{-1}\mathcal{R}||_x < 1$.
Due to $\mathcal{A}_D$ being diagonal, the Jacobi method applied to $\mathcal{A}$ corresponds to the splitting $\mathcal{A} = \mathcal{Z} - \mathcal{R}$, with $\mathcal{R}$ as before. The iteration matrix is $S_{Jac} = \mathcal{Z}^{-1}\mathcal{R}$. For any vector e we have:

$$||S_{Jac}e||_x \leq ||\frac{1}{\lambda}\mathcal{Z}_M^{-1}\mathcal{R}e||_x \leq \frac{1}{\lambda}||\mathcal{Z}_M^{-1}\mathcal{R}||_x||e||_x < \frac{1}{\lambda}||e||_x. \tag{3.31}$$

Hence, $\sigma \geq 1 - \frac{1}{\lambda}$. From $\lambda > 1$ we have $0 < \sigma \leq 1$. □

The Jacobi method corresponds to an incomplete factorization with a smaller level of fill than ILU(0): the diagonal. Since $(V + W)B$ is an M-matrix, from Manteuffel [68] (Theorem 3.6) we therefore have that the spectral radius of the ILU(0) operator has to be at least as small as the one of the Jacobi operator. This theorem does not require the factorized matrix to be symmetric. Consequently, $\rho(S_{VWB})$ is not only smaller than one but also smaller than $1-\sigma_{VWB}$, where σ_{VWB} depends on the 'weakest' diagonal dominance of B (and, hence, $(V + W)B$). As we have seen in Section 2.3, this is determined by the time step size but not by the problem size.

3.3.3.2 Convergence for the Pressure Part

We are now going to discuss the spectral radius of $S_A M_{AMG}$, where S_A results from the ILU(0) method for the linear system that is described by A and M_{AMG} describes AMG for this system. We note that, due to S_A and M_{AMG} not commuting, we cannot simply multiply the two spectral radii to compute $\rho(S_A M_{AMG})$. Instead, we will exploit that both solver operations do converge in the same norm. We will use the energy norm $||.||_A$, as it is the natural norm to be considered in the AMG context. It is defined via the scalar product $< A., . >_{std}$.

Since A is an spd M-matrix, AMG is applicable without any problems and we can expect it to be an efficient solver. We refer to the discussion by Ruge and Stüben [85, 95] for further details and proofs. We here just state that AMG is scalable for a class of problems at hand, e.g. A. That is, we have some $0 < \tau_A < 1$ such that $||M_{AMG}e||_A < (1 - \tau_A)||e||_A$ for an arbitrary error vector e, where τ_A does not depend on the problem size.

We now need to consider S_A and show that it does not have a negative effect. S_A results from the ILU(0) factorization of A, which implies $A = Q_A - R_A$ and $S_A = Q_A^{-1}R_A$, where $Q_A = L_A U_A$. Due to A being a symmetric M-matrix, from Meijerink and van der Vorst [71] we know that all entries of Q_A^{-1} and R_A are non-negative. That is, we have a regular splitting, which for our positive definite A implies [100] $\rho(S_A) < 1$. This implies that there is some consistent matrix norm such that $||S_A||_x < 1$. Also in the energy norm, due to the equivalence of norms in a finite dimensional vector space, the asymptotic convergence is guaranteed. However, we require $||S_A e||_A < ||e||_A$. Showing this requires the following preparations.

Lemma 3.14. *With ILU(0) on a symmetric non-zero pattern we have $Q_A = Q_A^T$ (in fact, the argumentation holds with ILU(k) as well).*

Proof. Since A is spd, we can compute the incomplete cholesky (IC) decomposition $A = G_A G_A^T - R_{AG}$ on a symmetric non-zero pattern, with G_A a lower triangular matrix. With $L_A = G_A diag(G_A)^{-1}$ and $U_A = diag(G_A)G_A^T$, we can turn this decomposition into an incomplete LU factorization with the same non-zero pattern. According to Lemma 3.7, this must be unique. Hence, the ILU(0) decomposition and the IC(0) one are the same. This gives $Q_A = L_A U_A = G_A G_A^T$ is symmetric. □

Lemma 3.15. *Let $S_A^{sym} = \frac{1}{2}(S_A + S_A^T)$ be the symmetric part of S_A. We have $\rho(S_A^{sym}) < 1$.*

Proof. See Appendix B. □

Remark 3.16. *We note that, although A and Q_A are symmetric, $S_A = Q_A^{-1}R_A$ generally is not symmetric. However, $AS_A = A(\mathbb{1} - Q_A^{-1}A)$ is symmetric, which implies that S_A is symmetric regarding the energy scalar product $< A., . >_{std}$. This is sufficient to, for instance, use ILU(0) as a preconditioner with the conjugate gradient method.* △

Corollary 3.17. *Since $\rho(S_A) < 1$, so is $\rho(S_A^2)$. With the same argumentation as in Lemma 3.15 we have $\rho(\frac{1}{2}(S_A^2 + (S_A^T)^2)) < 1$.* △

We are now ready to show our desired result.

Theorem 3.18. *For all e, we have $||S_A e||_A < ||e||_A$.*

Proof. The statement to be proven is equivalent to

$$0 < A - S_A^T A S_A.$$

Due to Q_A being symmetric we have $S_A^T = \mathbb{1} - A Q_A^{-1}$. Hence, by a multiplication of A^{-1} from the left, we can turn the above relation into

$$0 < \mathbb{1} - S_A^2.$$

That is, to show the theorem, we need to show that $\mathbb{1} - S_A^2$ is positive definite in a non-symmetric sense (we remember that S_A does not need to be symmetric). More precisely, it is sufficient to show that the symmetric part of $\mathbb{1} - S_A^2$ is positive definite[8]. We define $T := \frac{1}{2}(S_A^2 + (S_A^2)^T)$ and our requirement is:

$$0 < \mathbb{1} - T.$$

Let w be an eigenvector of T with eigenvalue λ. Then we have

$$w^T(\mathbb{1} - T)w = (1 - \lambda)w^T w,$$

which is positive: Due to $\rho(T) < 1$ (cf. Corollary 3.17), $\lambda < 1$. As T is symmetric, its eigenvectors form a basis of the $\mathbb{R}^m$ and, hence, $\mathbb{1} - T$ is spd. □

Corollary 3.19. *$\rho(S_A M_{AMG}) < 1 - \tau_A$ with $0 < \tau_A \leq 1$.*

Proof. With an arbitrary e we have $||S_A M_{AMG} e||_A < ||M_{AMG} e||_A < (1 - \tau_A)||e||_A$. □

3.3.4 Meaningfulness for the Initial Problem

In the previous sections we have discussed the convergence properties for the limit case where the pressure- and saturation-related sub-blocks of the linear system are related by a scaling with a positive, diagonal matrix. In the general Dead-Oil case, we can still expect these sub-blocks, i.e., A and $\hat{A}$ as well as B and $\hat{B}$, to be related. We have also outlined that we can even expect this linear dependency

[8] Let $\mathcal{A}$ be an arbitrary square matrix. Then, due to the Euclidean scalar product being symmetric, for any vector v there is: $\frac{1}{2}v^T(\mathcal{A} + \mathcal{A}^T)v = \frac{1}{2}(< v, \mathcal{A}v >_{std} + < \mathcal{A}v, v >_{std}) =< v, \mathcal{A}v >_{std} = v^T \mathcal{A} v$.

to dominate this relation. However, we can generally not expect it to be the only dependency. That is, we generally have

$$\hat{A} = VA + E, \tag{3.32}$$

with a positive, diagonal V and some rest matrix E. An analog relation holds with B and $\hat{B}$. However, the pressure sub-blocks are the crucial ones regarding the Schur complement (3.26) that ILU incorporates ($E = 0$) or approximates ($E \neq 0$). In Section 3.3.2, we have used the relation $\hat{B} = WB$ only to describe the ILU factorization of J_{limit} in Lemma 3.8.

That ILU approximates this Schur complement also in the $E \neq 0$ case is clear from the fact that the LU-decomposition (3.24) for the 2×2 unknown-wise system exactly corresponds to this Schur complement: ILU is an approximation of the full LU-decomposition.

Clearly, the quality of this approximation is the better, the "smaller" E is. This also determines how close the spectral radius of our general operator M is to the one of $\tilde{M}$ in the Schur complement case. That is, we can expect our System-AMG approach to work the better, the closer the pressure relations of the different involved phases are.

This is also reasonable in a physical interpretation: the pressure approximation from the coarse grid correction process only reflects the flow-properties regarding the first phase, which are described by A_{pp}, or A. Once the entire problem is solved, we clearly have a unique pressure that fits to the pressure relations from the second phase as well, which are described by A_{sp}, or $\hat{A}$. However, during the iterative solution process, we compute a pressure correction under a still erroneous saturation. Therefore, a pressure correction that, under this saturation, properly reflects the pressure relations from A, under this saturation, does not need to be well-suited regarding the pressure relations from $\hat{A}$. Both pressure relations may well differ, for instance, due to different pressure dependencies of densities. Hence, if $E \neq 0$, the ILU sweep will have to adopt the pressure accordingly and does not only need to update the saturations. The "bigger" E, the more impacts of these additional efforts can be expected on the overall convergence rate. We will get back to this aspect when constructing a preparatory matrix transformation in Chapter 4.

In summary, we have the following conditions for our System-AMG approach to be an efficient solver method:

- A is well-suited for the application of AMG.
- A and $\hat{A}$ are sufficiently similar, which means E from (3.32) is small.
- B and $\hat{B}$ are diagonally dominant in each row.

3.3.5 Empirical Confirmation

In order to confirm our results in practice, we are now going to discuss the relation of the pressure-related sub-parts with exemplary test cases. In Figure 3.5 we illustrate the small impact of the rest matrix E by comparing the Frobenius norms of A, $\hat{A}$ and E of a linear system from an incompressible SPE10 simulation.

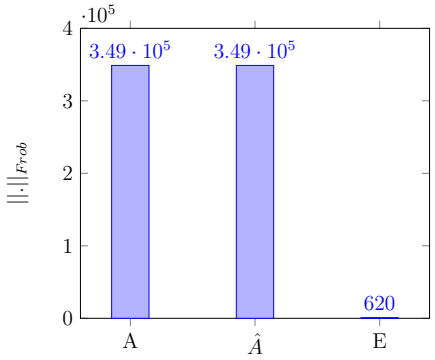


FIGURE 3.5: Frobenius norms of the matrices A, $\hat{A}$ (corresponding to A_{pp} and A_{sp}, respectively) and the rest matrix E from a linear system from an incompressible Dead-Oil simulation (SPE10 model).

A small E implies that V is a good approximation of $\hat{A}A^{-1}$, which implies that ILU for the full system is a good approximation of the Schur-complement (3.26). The good approximation of this Schur complement coincides with practical experiences. Let us construct the ILU(0) factorization $J = LU - R$ of a representative Jacobian from a Dead-Oil simulation and compute the Frobenius-norms of the four sub-matrices of J and $L^{-1}U^{-1}J$, respectively. As this requires explicitly inverting matrices and dense matrix computations, we do not consider the well-known SPE10 case here (1.1 million cells), but the much smaller Case10 from ADGPRS' test suite (9026 cells). For this representative problem, the norms of the sub-matrices are:

$$\text{Blocks in } J: \begin{pmatrix} 1.38 \cdot 10^5 & 1.11 \cdot 10^9 \\ 7.85 \cdot 10^4 & 1.11 \cdot 10^9 \end{pmatrix} \quad \text{Blocks in } L^{-1}U^{-1}J: \begin{pmatrix} 95.36 & 6.57 \cdot 10^5 \\ 0.7 \cdot 10^{-4} & 94.91 \end{pmatrix}.$$

The small lower left block of $L^{-1}U^{-1}J$, especially in contrast to the upper right block, shows the good approximation of the Schur complement by ILU(0).

The fact that the Schur complement is only approximated, however, implies that the convergence of System-AMG may depend on the size of E. In fact, the convergence speed for two SPE10 problems at different mesh sizes (original size and refined) is not exactly equal, as depicted in Figure 3.6.

However, we still see the clear advantage of AMG: For both sizes, the convergence

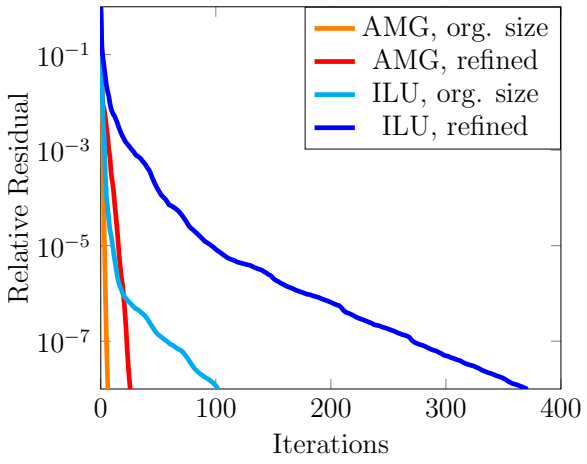


FIGURE 3.6: Convergence history of System-AMG and ILU(0) for two linear systems from Dead-Oil simulations with different mesh sizes. One simulation is the original SPE10 with 1.1 million cells, the other one is a refined SPE10 model with nearly 9 million cells (cf. Appendix A). Both linear systems are the first system from a simulation run with ADGRPS.

of ILU starts slowing down once a certain accuracy is achieved. This is different for the solution with System-AMG. Here we have roughly the same residual reduction per iteration, which is bounded away from one. This implies that System-AMG reduces all high- and low-frequency error components sufficiently well per iteration, while the convergence of ILU suffers from some error components that are not reduced properly.

3.3.6 Meaningfulness for a Full Black-Oil Problem

We have made some assumptions regarding our model problem (3.21) and we are now going to discuss the impact of these assumptions not to hold in practical applications. Essentially, all assumptions have been made in order to ensure the

applicability of the AMG and ILU theory regarding A.
If, where MPFA discretizations are used, A is of essentially positive type, the AMG theory still is applicable. This, however, does not hold for the ILU theory.
If we allowed for compressible models, according to Theorem 2.25, A still was an M-matrix, but not symmetric anymore. The M-matrix property at least holds under certain constraints on the wells and the time step. From practical experience, we therefore still expect AMG to be applicable. However, no statements regarding an energy norm, which does not exist for a non-symmetric matrix, are possible. The same holds for ILU.
Finally, also the consideration of only two phases is a limitation of generality. However, we expect our discussion to also hold with more phases being involved. Then A_{ss} is not related to a single physical unknown anymore, but to two or more. However, it remains strongly (block) diagonally dominant with positive diagonals. Just, due to A_{ss} being a 2×2 block-matrix with the blocks being Z-matrices, A_{ss} is not an M-matrix anymore and we cannot apply the ILU theory. From experience we expect ILU to work properly, though. Also, all sub-blocks of A_{sp} still are comparable to A_{pp}, as they result from the same background.

Chapter 4

System-AMG for Industrial FIM Black-Oil Simulations

In the previous chapter we have defined a basic System-AMG approach for Black-Oil simulations under somewhat ideal conditions. However, for industrial models these ideal conditions are hardly ever met. In Chapter 2 we have seen various examples for model and material properties that may have strong influences on the linear systems and the solver performance. Especially well settings under real conditions are becoming increasingly complex with today's well-bore capabilities. In Section 2.4.1.2 we have seen that the source terms from wells can have a drastic impact on the properties of the pressure related matrix parts. This may even result in these matrix parts being indefinite. As the well-settings might change over time, also the amount of such impacts can change in the scope of a simulation run.

In this chapter we will discuss a matrix transformation that we can apply to the original linear systems in order to maintain the applicability of our basic System-AMG approach for a wide range of Black-Oil problems. In fact, the method allows to robustly and efficiently solve all problems that we have been concerned with. More precisely, we transform the input problem into an equivalent linear system by a matrix scaling with non-singular matrices C_L and C_R:

$$Ju = f \iff \tilde{J}y = g \text{ with } \tilde{J} = C_L J C_R,\ y = C_R^{-1}u \text{ and } g = C_L f. \tag{4.1}$$

In the following a tilde ˜ indicates a scaled matrix, or a sub-part of a scaled matrix.

In this chapter we will discuss a left scaling C_L that allows to solve Black-Oil problems robustly with System-AMG. Our discussion on how to extent these ideas to more complex simulations in the next chapter will also involve right scalings.

4.1 General Aspects of Matrix Transformations

We apply the matrix transformation to linear systems that are set up by a reservoir simulator. Typically, these are ordered point-wise (cf. (2.2)). We call $[J]_{ij}$ the block that couples the unknowns of grid cell i with those of cell j. The size of these blocks is determined by the number of physical unknowns that exist in the respective cells. Due to a disappearance of phases, this may vary between different cells.

We will only consider Black-Oil problems in this chapter and per point have to deal with three unknowns or less (one pressure and up to two saturations). We will extent our System-AMG approach to more complex simulations in Chapter 5. We may then have to deal with significantly more unknowns per point. Therefore, we directly discuss the matrix transformations with an arbitrary size of the blocks $[J]_{ii}$, rather than only size three.

Each block is a matrix by itself, reflecting the structure of the unknown-wise ordering (2.1). In the Black-Oil case an arbitrary block reads as:

$$[J]_{ij} = \begin{pmatrix} [A_{pp}]_{ij} & [A_{ps}]_{ij} \\ [A_{sp}]_{ij} & [A_{ss}]_{ij} \end{pmatrix}. \tag{4.2}$$

With the above notation we refer to the parts of the respective sub-matrices from the unknown-wise notation that correspond to the interaction between cells i and j. In the Black-Oil case $[A_{pp}]_{ij}$ is a scalar, $[A_{ss}]_{ij}$ is a 2×2 matrix and $[A_{sp}]_{ij}, [A_{ps}]_{ij}$ are 2×1 and 1×2 vectors, respectively.

Throughout this thesis, in the sense of this point-wise block structure, we only consider block-diagonal scalings. That is, with small[1], possibly dense blocks $[C_L]_i$ and $[C_R]_i$, we have

$$C_L = \begin{pmatrix} [C_L]_1 & & \\ & \ddots & \\ & & [C_L]_{n_{points}} \end{pmatrix} \text{ and } C_R = \begin{pmatrix} [C_R]_1 & & \\ & \ddots & \\ & & [C_R]_{n_{points}} \end{pmatrix}. \tag{4.3}$$

[1]The dimension of $[C_L]_i$ and $[C_R]_i$ corresponds to the dimension of $[J]_{ii}$.

By limiting the scalings to block-diagonal ones, we minimize their impact on the sparsity structure of the Jacobian (2.14). The application of an arbitrary left- or right scaling would require significantly more memory and computational efforts. For an efficient solver approach in reservoir simulations this was impractical, and it turns out to be not needed.

Clearly, solving the scaled problem in exact arithmetic gives the solution of the original problem $Ju = f$. However, we have to quickly review the impacts of solving a transformed linear system by an iterative method.

- With a left scaling we solve $C_L J\tilde{u} = C_L f$. In the i-th iteration, for the solution iterate, residual and error, the relation between the scaled and the original problem is:

$$u_i = \tilde{u}_i,\ e_i = \tilde{e}_i \text{ and } r_i = C_L \tilde{r}_i. \tag{4.4}$$

- With a right scaling we solve $JC_R\tilde{u} = f$. In the i-th iteration we have:

$$u_i = C_R\tilde{u}_i,\ e_i = C_R\tilde{e}_i \text{ and } r_i = \tilde{r}_i. \tag{4.5}$$

In a practical implementation, the accuracy of a solution iterate typically is measured in terms of the residual. In our discussion, for reasons of consistency, we will always consider the residual reduction that corresponds to the original problem, i.e., the residual is scaled back in the case of left scalings being involved.

4.2 Status Quo of CPR-AMG: Approximate Pressure-Saturation Decoupling

Left scalings of the Jacobian are a common practice in today's reservoir simulators when using CPR-AMG (cf. Section 3.2.2). The idea is to weaken the coupling between the pressure and saturation unknowns, i.e. A_{ps}. This way, the outer convergence of CPR-AMG shall be ensured, and accelerated as much as possible. However, from the viewpoint of AMG, there are serious drawbacks, as we will see in the following. In Section 4.4 we will discuss a new scaling that turns out to be more suited in this regard.

4.2.1 Purpose from CPR's Point of View

In CPR-AMG, Scalar-AMG is used to solve a problem that in fact is not an independent, scalar problem. The pressure unknown depends on further unknowns in a coupled system. It is common practice to, at least approximately, decouple the pressure from the saturations. This shall reduce the pressure correction step's dependency on the, in the iterative solution process still erroneous, saturation unknowns.
Let us consider the limit case of a full decoupling with a full Schur complement:

$$C_L J = \begin{pmatrix} \tilde{A}_{pp} & 0 \\ A_{sp} & A_{ss} \end{pmatrix} \text{ with } C_L = \begin{pmatrix} \mathbb{1} & -A_{ps}A_{ss}^{-1} \\ 0 & \mathbb{1} \end{pmatrix}. \tag{4.6}$$

Given that Scalar-AMG worked properly for $\tilde{A}_{pp}$, the solution approach would be straight forward: After having sufficiently solved for the pressure unknown, only updating the saturation unknowns remains to be done. Due to the diagonal dominance of A_{ss}, this is rather inexpensive.
Such a complete decoupling is only possible by computing the full Schur complement, i.e., including the exact inversion of A_{ss}. Since the blocks of A_{ss} are diagonally dominant but not diagonal, such a full decoupling is practically impossible. Therefore, pressure-saturation decouplings are performed only approximately in CPR approaches.

4.2.2 Approximate Decoupling Methods

There are various approximate decoupling methods reported in the reservoir simulation literature, as we have already outlined in Chapter 1. All of these methods aim at removing the pressure-saturation coupling in the diagonal blocks $[J]_{ii}$. In Section 2.3.2.1 we have seen that the saturation related matrix blocks A_{ps} and A_{ss} are strongly (block) diagonally dominant. Hence, the dominating coupling between pressure and saturation is removed this way. The effect of the scaling on the off-diagonal blocks is simply accepted 'as is'.
We do not give a full overview on all available methods here, but only review the most relevant ones.

4.2.2.1 Alternate Block Factorization

The *Alternate Block Factorization (ABF)* was initially introduced by Bank et.al. [9] to reduce intra-equation couplings in a general system of linear systems that, for instance, resulted from a system of PDEs. This is achieved by scaling J such that the diagonal blocks become the identity, i.e.

$$C_L = \begin{pmatrix} [J]_{1,1}^{-1} & & \\ & \ddots & \\ & & [J]_{n_{points},n_{points}}^{-1} \end{pmatrix}. \tag{4.7}$$

Although the method does not particularly focus on the pressure saturation coupling in the diagonal, it removes this coupling for all diagonal blocks $[J]_{ii}$.

4.2.2.2 Quasi-IMPES

The idea of *quasi-IMPES (qIMPES)* is to approximate the Schur complement from (4.6) by a Schur complement for the diagonals only. This corresponds to defining C_L as:

$$[C_L]_i = \begin{pmatrix} \mathbb{1} & -[A_{ps}]_{ii}[A_{ss}]_{ii}^{-1} \\ 0 & \mathbb{1} \end{pmatrix}. \tag{4.8}$$

The possibly remaining couplings between pressure and saturations in the off-diagonal blocks $[J]_{ij}$ are usually accepted. Alternatively, some simulators even simply drop these terms and, hence, do no longer solve a linear problem that is equivalent to the initial one.
A more detailed description of the approach is, for instance, given by Jiang or Lacroix et.al. [52, 63].

4.2.2.3 True-IMPES

The *true-IMPES (tIMPES)* approach incorporates physical information in the approximate decoupling. For the saturation related matrix blocks, it distinguishes between those parts that result from the flux part of the mass balances (2.10), and those from the accumulation part. The Schur complement approximation is only applied to the flux related coupling between pressure and saturation. That is, if we split $[A_{ps}]_{ii} := [A_{psFlux}]_{ii} + [A_{psAccu}]_{ii}$, then the approach aims at only removing

$[A_{psFlux}]_{ii}$ in the diagonal block $[J]_{ii}$ by a Schur complement. If we analogously subdivide $[A_{ss}]_{ii}$, C_L is defined as:

$$[C_L]_i = \begin{pmatrix} \mathbb{1} & -[A_{psFlux}]_{ii}[A_{ssFlux}]_{ii}^{-1} \\ 0 & \mathbb{1} \end{pmatrix}. \tag{4.9}$$

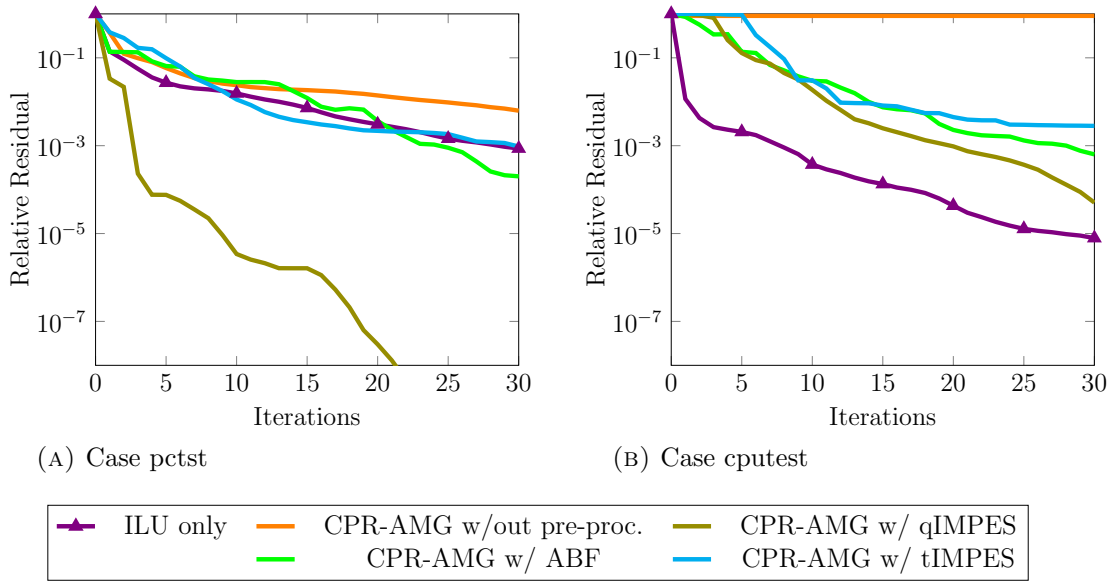
This way, the Schur complement only involves terms that are induced by the flux. This is generally assumed to be closer to a physical intuition (see, for instance, [52], section 7.4.1).
Just as with the qIMPES, remaining couplings between pressure and saturation are usually accepted and sometimes simply dropped. In contrast to qIMPES, such couplings may also remain in the diagonal blocks here. For further discussion on the approach, we refer to Jiang and Lacroix et.al. [52, 63].

Remark 4.1. *The application in a purely algebraic manner does not appear to be straight forward. The splitting between flux and accumulation induced terms of the matrix requires physical information. However, from our discussion in Section 2.3.2.1 it is clear that in off-diagonal blocks $[A_{ps}]_{ji}$ and $[A_{ss}]_{ji}$ entries are induced by the flux terms of the mass balance, only (cf. Equations (2.23) and (2.24)). The entries in the ji-blocks result from the derivatives of the transmissibilities w.r.t. the saturations in cell i (in contrast to ji, in the ij-block, it would be the saturation in cell j). Hence, the column-wise sum of the absolute values of these off-diagonal terms must equal the flux induced part of $[A_{ps}]_{ii}$ and $[A_{ss}]_{ii}$, respectively.* △

4.2.3 Results and Drawbacks for CPR-AMG

In practice, we always observe some simulations where CPR-AMG with the approximate decouplings fails. Such failures are observed with different matrices for the different approximate decouplings. Figure 4.1 shows two exemplary cases where we can see the different impact of the approximate decoupling methods. True-IMPES was constructed algebraically, as described in Remark 4.1. We compare the different solution approaches with FGMRes/ILU(0) as a reference benchmark that was able to sufficiently solve all given problems.
We will provide results with further test cases in Figure 4.4 in Section 4.4.3.

Such rather unpredictable failures of the CPR-AMG method are problematic for the application in reservoir simulators. These issues nearly always are related to



(A) Case pctst (B) Case cputest

FIGURE 4.1: Convergence history of the CPR-AMG approach with different approximate decoupling methods, compared to a single level incomplete factorization. Two representative systems from three dimensional, compressible Black-Oil simulations with 195193 cells in (A) and 29540 cells in (B) (cf. Appendix A).

problems that AMG has with the pressure sub-problem in the approximately decoupled system. These problems are a consequence of the fact that the objective of the methods from Section 4.2.2 is the approximate decoupling of pressure and saturation, but not the applicability of AMG. As all presented approximate decoupling methods result in some change in the pressure sub-problem, they clearly can have an impact on AMG.

More precisely, the left scaling mixes the different pressure related blocks that correspond to cell i according to:

$$[\tilde{A}_{pp}]_{ij} = \xi_{1,i}[A_{pp}]_{ij} + \begin{pmatrix} \xi_{2,i} \\ \vdots \\ \xi_{n_{unknowns},i} \end{pmatrix}^T [A_{sp}]_{ij}. \tag{4.10}$$

The weights $\xi_{k,i}$ might vary drastically between two blocks i_1 and i_2. Hence, before summing them up, each of the blocks (A_{pp} and the blocks in A_{sp}) are scaled by some diagonal matrices. This scaling may differ for each cell and each unknown. It is computed only with the objective to approximately decouple pressure and saturations and does not at all reflect the properties of the resulting $\tilde{A}_{pp}$. $\tilde{A}_{pp}$ is

not only significantly non-symmetric in general, but may also be indefinite.
Both effects can have drastic impacts on AMG's applicability. A strong non-symmetry of the problem is rather not natural for a diffusion-based problem[2]. It is questionable whether the construction of the AMG coarsening allows for a reasonable interpolation. This is because in a non-symmetric problem, a matrix coupling a_{ij} may be considered strong according to (3.7), while a_{ji} is not. In any case, AMG cannot construct a hierarchy that properly reflects the diffusion effect, as the matrix properties that result from the diffusion are mixed with completely different parts. Finally, a possibly resulting indefiniteness, as it is observed in practice, may cause even more severe convergence issues for an AMG method.

We note that any (approximate) pressure-saturation decoupling for a general model will have some impacts on the pressure sub-system, if we used a left-scaling for the decoupling.
A right-scaling cannot be used as an alternative: While this could be constructed without changing the pressure sub-problem, it would result in saturation columns being influenced by the pressure column. As the pressure-related sub-matrices in general are only weakly diagonally dominant, we have to expect quite some influence on the off-diagonal pressure-saturation coupling, if we only decoupled the diagonal. A full decoupling, also from the right, is not realizable in an efficient manner.
We have seen that CPR-AMG and System-AMG for Black-Oil problems are nearly identical methods. Hence, if we used the approximate decoupling methods prior to the application of System-AMG, we would have to expect the same impacts as in CPR-AMG. Therefore, we do not consider (approximate) pressure-saturation decouplings.

Remark 4.2. *In addition to unpredictable problems for the applicability of AMG, there is another problem with approximate decouplings: They only aim at removing the initially dominating term of the coupling between pressure and saturation. The effect on the respective couplings in off-diagonal blocks is unpredictable. Hence, it is not clear whether decoupling pressure and saturation in the diagonals really always 'weakens' the respective overall coupling, i.e., whether* $||\tilde{A}_{ps}|| < ||A_{ps}||$ *in some matrix norm.* △

[2]In contrast to this, the non-symmetry of the pressure problem in compressible simulations (cf. Theorem 2.25) is typically rather small, and the symmetric influences from the fluid flux dominate.

4.3 Challenging Initial Linear Systems

In Figure 4.1 from the previous section we have not only seen that approximate decoupling methods may have some problematic effects from AMG's point of view, we have also seen that CPR-AMG without any matrix pre-processing might not work as well. Consider the second example in the figure, the test case cputest: Without any pre-processing CPR-AMG does not converge at all, which holds for our System-AMG approach from Figure 3.2 as well. There are also several other test cases observed where System-AMG does not work for the linear problem "as is".

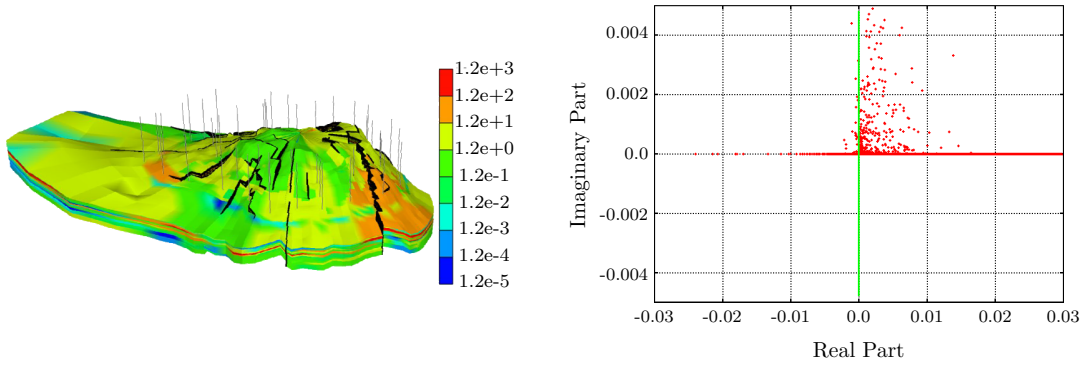
The reason are problems with the pressure-based AMG hierarchy. When we discussed our System-AMG approach in Chapter 3, we assumed somewhat mild conditions that especially ensured the pressure related matrix blocks to be M-matrices, i.e., in particular to be definite. However, these conditions do not need to (and typically do not) hold in practice. Whether or not we need to expect negative impacts on the applicability of AMG, depends on the model and especially the used wells:

- In incompressible models, according to Theorem 2.21, we always have $A_{pp} > 0$ being an M-matrix, if we used TPFA discretizations. Following our discussion in Section 2.3.2.2, by using MPFA schemes we can still expect A_{pp} to be of essentially positive type. In both cases A_{pp} meets the requirements of AMG.
- In compressible models, however, more physical details are involved, making them more realistic. Following Theorem 2.25 and the discussion beforehand, for larger time steps some rows of A_{pp} may no longer be even only weakly diagonally dominant. This does not necessarily cause problems for AMG, however, it may. Most importantly, however, in compressible models, wells can cause serious problems for AMG's applicability (cf. Section 2.4.1.2):
  - Injection wells, depending on how strongly the density of a phase depends on the pressure, can result in negative contributions to the diagonals of A_{pp}. In practice, these can have a serious impact on AMG. They can even be so strong that the entire diagonal becomes negative.
  - Production wells yield positive contributions to the matrix diagonal. This is not a problem for AMG.

To formalize these findings, for a compressible model let us denote the pressure matrix that we would have obtained in the incompressible case as A_{pp}^{incmp}. According to our discussion in Section 2.3.2.1 we have

$$A_{pp} = A_{pp}^{incmp} + A_{pp}^{rest}. \tag{4.11}$$

While A_{pp}^{incmp} results from the diffusion process, the rest matrix results from the compressibility. This rest matrix is non-symmetric and may have strong negative diagonal entries, possibly causing difficulties for AMG. A_{pp}^{incmp} meets AMG's requirements: Following Theorem 2.21, as long as each part of the reservoir is perforated by at least one well, A_{pp}^{incmp} is an spd M-matrix (or of essentially positive type in the case of MPFA discretization schemes). In the incompressible case we simply have $A_{pp}^{rest} = 0$.



(A) Permeability Field (B) Excerpt of Eigenvalue Plot

FIGURE 4.2: (A) Permeability field (in millidarcy) of the cputest case, with sketched wells (cf. Appendix A). Picture from Computer Modeling Group Ltd., CMG.
(B) Excerpt of the eigenvalues of a representative, original pressure sub-problem A_{pp} from cputest, plotted in the complex plane. Excerpt shows all eigenvalues with negative real part, but not all with positive one.

Especially if the pressure sub-problem is (highly) indefinite, the ideas behind the construction of an AMG hierarchy do not robustly apply anymore. As an example, consider the cputest case from an industrial reservoir simulation from CMG with their IMEX simulator (cf. Appendix A). We have already seen that AMG has convergence problems in Figure 4.1. This model is characterized by a high number of injection wells, where gas and water is injected. In the considered exemplary linear system, the impact of A_{pp}^{rest} on the pressure sub-problem is serious: there are 11% of the diagonals being negative. Roughly 2% of the eigenvalues of A_{pp} have a

negative real part[3]. Figure 4.2b shows an excerpt of the eigenvalues to illustrate this fact. Figure 4.2a shows the respective permeability field and the relatively high density of wells.
To illustrate the difficulties that System-AMG has when using A_{pp} as a primary matrix to construct the hierarchy with, let us consider Scalar-AMG being applied to the scalar pressure system. If we used standard[4] Scalar-AMG as a preconditioner for FGMRes, the solver does not converge at all. In fact, a massive divergence is only prevented by FGMRes.

Remark 4.3. *For this particular scalar problem it is still possible to construct a convergent AMG method:*

- *We use an alternating Schwarz approach that separates those rows with negative pressure diagonals and uses a direct solver for them (see, for instance, Gander [44]. Clees and Ganzer [27] also suggest Schwarz methods, for instance, for well equations).*
- *The coarsening is based on aggregation (cf. Remark 3.2). This allows to easily introduce an additional constraint in the coarsening: An aggregate is constructed only if the resulting row of the coarse level operator had a positive diagonal. Hence, the next level did not have negative diagonals.*

With these remedies, AMG converges robustly for the pressure sub-problem. This also holds for the full problem. However, the convergence rate is relatively poor and it is questionable whether this approach still maintains convergence with the next problematic case. Our goal is a System-AMG approach without such "manual adjustments". △

4.4 Dynamic RowSum Transformation for System-AMG

In this section we are going to discuss a matrix transformation that we will combine with the System-AMG approach to obtain a robust solver for Black-Oil problems,

[3]Due to the non-symmetry of this pressure sub-problem from a compressible simulation (cf. Theorem 2.25), there are not only real eigenvalues.

[4]The rows with negative diagonals are formally multiplied with -1 in order to have all diagonals positive for AMG.

also in complex industrial simulations. We call this transformation *dynamic row-sum*, or DRS. This transformation has three objectives:

- We will ensure that the matrix transformation does not introduce artificial difficulties from AMG's point of view. This was different with the approximate decouplings from Section 4.2.2.
- We will attempt to ensure matrix properties that we expect AMG to be well-suited for. That is, we will shield AMG from some of the negative impacts that may result from the complex physics in industrial simulations.
- We will attempt to improve the overall convergence, where this can be achieved without impacts on the first two objectives. More precisely, we compute a correction for the total pressure, rather than the primary phase's one, where this does not have negative effects on the applicability of AMG.

Remark 4.4. *In adaptive implicit simulations, for reasons of computational efficiency, there still is a need to decouple pressure and saturations. In Section 6.2 we will discuss this aspect and describe transformation approaches that can exploit the matrix properties in AIM simulations without the risk of loosing the applicability of AMG.* △

4.4.1 Full RowSum Transformation: Total Pressure Correction

The choice of associating the oil phase's mass balance equation with the pressure unknown for constructing the Jacobian matrix in Section 2.3.2 was completely arbitrary. As we observed in Corollaries 2.22 and 2.26, the background of A_{pp} and the blocks within A_{sp} is the same. All result from taking a mass balance equation's derivative w.r.t. pressure. Consequently, we could equally well have any other of the mass balances associated with the pressure. However, while the structure is roughly the same, the concrete material properties might vary per phase. Hence, the different pressure related blocks are not exactly equal.
If we recall the relation (3.32) from Section 3.3.4, with some diagonal matrices $V_1, V_2 > 0$ and some matrices E_1, E_2 we have:

$$A_{sp} = \begin{pmatrix} V_1 A_{pp} + E_1 \\ V_2 A_{pp} + E_2 \end{pmatrix}. \tag{4.12}$$

Given that AMG works properly for A_{pp}, we have seen that we can expect the convergence of System-AMG to be the better, the smaller E_1 and E_2 are - if we assume the Dead-Oil result to carry over to the Black-Oil case.
Due to the AMG hierarchy being constructed based on A_{pp}, it only reflects the coupling structure of the pressure that results from the first phase. Hence, under the still erroneous saturation in the iterative process, the resulting pressure correction does not necessarily reflect the respective structures that result from the other phases.
In contrast to this, in an IMPES approach from Section 2.3.1, a total pressure correction was considered. That is, an averaged pressure description (2.12), weighted with the current saturations was used. At the algebraic linear solver level, we do not have the saturations available. However, we could still use an averaged pressure description, with the same weight for each phase.
This corresponds to the transformation:

$$[C_L]_i = \begin{pmatrix} 1 & 1 & \dots & 1 \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}. \tag{4.13}$$

We call this *full row sum*, or FRS method. From the discussion above, we can expect it to be beneficial for the overall convergence of our System-AMG approach. Such a scaling was also proposed by Scheichl et.al. [90]. It was designed to approximately decouple pressure and saturations. Motivated from the fact that A_{ss} results from the second and third mass balance equations' dependency on the saturations, and A_{ps} from the dependency of the first mass balance on "$(1-S)$", a certain reduction of the pressure-saturation coupling can be expected (cf. the relation between B and $\hat{B}$ in the Dead-Oil problem J_{model} from Section 3.3). The strength of this reduction depends on how equally distributed the phases are, i.e., the difference of the three saturations, and on how non-linear the dependency of the mass-balances on the saturations is.
In contrast to the approximate pressure-saturation decouplings from Section 4.2.2, the FRS method does not introduce additional difficulties from AMG's perspective. If A_{pp} and the blocks in A_{sp} are positive definite, then this also holds for $\tilde{A}_{pp}$, which is the sum of all blocks.

4.4.2 Dynamic Weights

While the FRS method does not introduce new difficulties from AMG's perspective, the original problem may already have had some properties that are problematic for the robust application of AMG, as we have discussed in Section 4.3. In order to shield AMG from such properties, we will use dynamic summation weights that we can adjust according to (expected) problematic matrix properties. We will also further adjust the dynamic weights, depending on whether or not we expect a phase to give a meaningful contribution in the sense of a total pressure equation.
Such dynamic weights $\delta_j^i \in \mathbb{R}$, for block i lead to the transformation

$$[C_L]_i = \begin{pmatrix} \delta_1^i & \delta_2^i & \dots & \delta_{n_{unknowns}}^i \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}. \tag{4.14}$$

The previous FRS scaling is the special case where $\delta_j^i = 1$ for all j and i.

4.4.2.1 Shielding AMG from Problematic Pressure Problems

Let us start our discussion with the primary objective: the applicability of AMG. In Theorem 2.25 we have seen that we cannot necessarily expect A_{pp} to be perfectly suited for AMG. It may especially become indefinite, which may result in serious convergence issues for AMG. As a matter of practical experience, an AMG method applied to such a problem can diverge drastically.
Let us recall $A_{pp} = A_{pp}^{incmp} + A_{pp}^{rest}$ from (4.11), with A_{pp}^{incmp} being symmetric and positive definite. In compressible simulations with injection wells, some diagonals of A_{pp}^{rest} may be strongly negative. That is, while A_{pp}^{incmp} always is at least weakly diagonally dominant, due to A_{pp}^{rest}, this does not need to hold for A_{pp}.
There is a clear relation between the induced lack of diagonal dominance and indefiniteness of A_{pp}. We can illustrate this with two artificial cases:

- Let λ be an eigenvalue of A_{pp}^{incmp} and $\epsilon > 0$. Then, with $A_{pp}^{rest} = -(\lambda + \epsilon)\mathbb{1}$, A_{pp} is indefinite.

- An injection well does not cause all diagonals of A_{pp}^{rest} to be negative. Let us consider the case $A_{pp}^{rest} = -\eta\, DIAG(e_k)$, where e_k denotes the k-th Euclidean basis vector and $DIAG(e_k)$ the diagonal matrix where the i-th diagonal is the i-th entry of e_k. With an eigenvalue λ of A_{pp}^{incmp} and the corresponding eigenvector v, A_{pp} is indefinite as soon as $\eta v_k^2 > \lambda v^T v$. That is, η may be greater than the smallest eigenvalue of A_{pp}^{incmp} without causing indefiniteness in A_{pp}. However, with η big enough, A_{pp} is indefinite.

It is not predictable how strongly the diagonal dominance of A_{pp} in a particular row needs to be violated to have an indefinite A_{pp}, at least not in an efficient way. Moreover, there typically also are diagonally dominant rows in A_{pp}. Production wells can only improve the diagonal dominance of the corresponding matrix rows. Also injection wells do not necessarily result in a violation of the diagonal dominance. Following our discussion in Section 2.4.1.2, an injection well in a compressible model has a positive and a negative contribution to the diagonal. The effect on the diagonal dominance depends on which of both dominates. However, a single injection well may already be sufficient to cause indefiniteness.

We have seen that the pressure parts of the Jacobian, i.e., A_{pp} and the blocks in A_{sp}, do not need to be equal. This especially holds for the rest matrices they include. That is, all these blocks may suffer from properties that AMG might have problems with. However, these problematic matrix properties result from pressure derivatives of the phases' densities. These differ for the different phases and we can expect the extent and the location of these problems to be different for the different pressure related blocks.
We are going to define our summation weights to take this fact into account. We first of all exclude rows from the summation process that have a negative pressure diagonal. Moreover, we exclude those rows where the pressure sub-part suffers from a "too drastic" lack of diagonal dominance. With some control parameter $0 \leq \epsilon_{DD} \leq 1$, we define the summation weights as:

$$\delta_j^i := \begin{cases} 0 & \text{for } [a_{j,1}]_{i,i} < 0 \\ 0 & \text{for } \frac{|[a_{j,1}]_{i,i}|}{\sum_{k \neq i} |[a_{j,1}]_{i,k}|} < \epsilon_{DD} \\ 1 & \text{else} \end{cases} . \tag{4.15}$$

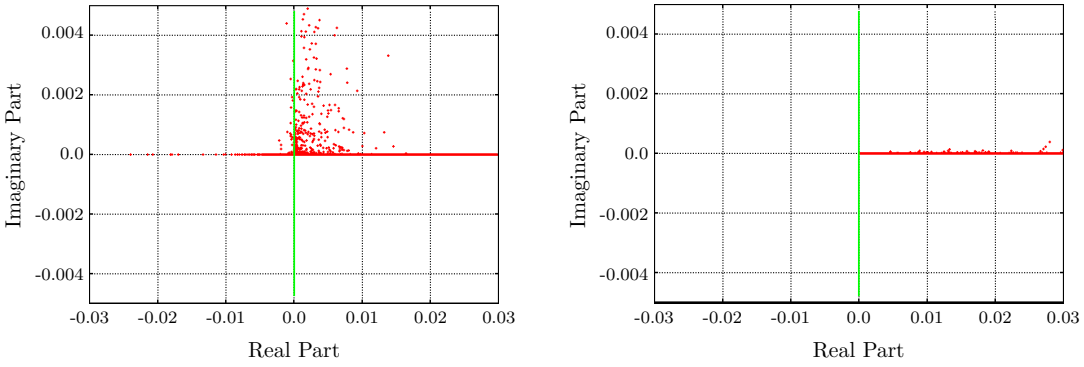
With the parameter ϵ_{DD}, we can control the strictness in requiring diagonal dominance. We see this by discussing the two limit cases regarding the control parameter. For this purpose, let us assume that for each block-row i we find at least one k such that $\delta_k^i \neq 0$. We will discuss what to do if this assumption should not hold further below.

- With $\epsilon_{DD} = 1$ we exclude all rows where the diagonal dominance is even only slightly violated. Then the resulting pressure sub-block $\tilde{A}_{pp}$ is at least weakly diagonally dominant in each row. Hence, we can expect it to be absolutely well-suited for AMG.
  However, we ignore all rows with only a minor lack of pressure related diagonal dominance. As we have seen, a small lack of diagonal dominance does not necessarily cause indefiniteness. Hence, we possibly ignore more rows than necessary to avoid problems for AMG. The beneficial effect from computing a total pressure is weakened more than necessary. That is, the convergence of System-AMG might be better with a smaller ϵ_{DD}.
- With $\epsilon_{DD} = 0$ only rows with a negative pressure-diagonal are excluded. Thus, we typically are much closer to generating a total pressure problem in $\tilde{A}_{pp}$. However, as there might still be a drastic lack of diagonal dominance, the resulting pressure sub-problem may be indefinite, with all possible implications on AMG's applicability.

With the choice of ϵ_{DD} we need to find a trade-off between the robust applicability of the System-AMG approach, with a maybe slow overall convergence, and the acceleration of the System-AMG convergence.
We can see the beneficial effect of the DRS transformation with the cputest case that we have discussed before (cf. Figure 4.2): Figure 4.3 shows an excerpt of the eigenvalues of the pressure sub-problem before and after the application of the DRS transformation (i.e., A_{pp} and $\tilde{A}_{pp}$, respectively). For this particular problem, no eigenvalues with negative real part remain. We can robustly solve the problem with System-AMG, as we are going to see in Figure 4.4 later on.

4.4.2.2 Ignoring Phases of Minor Influence

The DRS method does not primarily consider the coupling between pressure and saturation in A_{ps}. To a certain extent, however, we can monitor this in addition. More precisely, we check whether in A_{ps} the coupling between the pressure and



(A) Original A_{pp}. (B) $\tilde{A}_{pp}$ after DRS.

FIGURE 4.3: (A) Excerpt of the eigenvalues of the original pressure sub-problem A_{pp} from cputest, plotted in the complex plane. Excerpt shows all eigenvalues with negative real part, but not all with positive one. (cf. Figure 4.2)
(B) Excerpt of the eigenvalues of the pressure sub-problem $\tilde{A}_{pp}$ after the application of DRS.

any of the saturations already is small. If, with some control parameter $\epsilon_{ps} > 0$, we have a $k > 1$ such that

$$\frac{\sum_{j \neq i} |[a_{1,k}]_{i,j}|}{|[a_{1,1}]_{i,i}|} < \epsilon_{ps}, \tag{4.16}$$

then we exclude the respective row from the summation by setting $\delta_k^i = 0$. However, we only apply this criterion if, after the first step, δ_1^i still is one, i.e., the first phase's pressure part still is part of $\tilde{A}_{pp}$.

We do so, because the small coupling in A_{ps} implies that the pressure problem that is based on the first phase, i.e., the problem described by A_{pp}, sufficiently well describes the pressure problem that is associated with the k-th phase.

Consider the extreme case of $A_{ps} = 0$. Then the pressure that solves the problem described by A_{pp} is the pressure solution, independent of whether the saturations already are correct. In this sense, the linear system that is described by A_{pp} properly captures the pressure problems that result from the other phases.

In general we do not have $A_{ps} = 0$. However, regarding our phase k we assume the pressure description that results from the first phase to be sufficient. Involving it in the averaging process does not give any contribution in terms of the total pressure description, but just adds some 'noise'.

4.4.2.3 Regularity of the Transformation

In order to obtain a reasonable transformation, we need to ensure that C_L is non-singular, which is equivalent with each block $[C_L]_i$ being non-singular. After the dynamic summation, a singularity may occur if $\delta_1^i = 0$ for any i. Then the first column of $[C_L]_i$ was completely filled with zeros and the block was singular.
In the easiest case there is some $k > 1$ with $\delta_k^i \neq 0$. Then we simply adjust the k-th row of $[C_L]_i$ and obtain:

$$[C_L]_i = \begin{pmatrix} 0 & \delta_2^i & \dots & \delta_k^i & \dots & \delta_{n_{unknowns}}^i \\ 0 & 1 & & & & \\ \vdots & & \ddots & & & \\ 1 & & & 0 & & \\ \vdots & & & & \ddots & \\ 0 & & & & & 1 \end{pmatrix}. \tag{4.17}$$

Because $\delta_k^i \neq 0$, the above block is non-singular without further implications for the resulting pressure sub-problem. This adjustment in fact corresponds to an exchange of rows. That is, instead of associating the pressure with the primary phase α, for cell i we are associating it with the k-th phase. This is reasonable, as the initial choice α was completely arbitrary. Hence, we are just choosing another association that causes less problems for AMG.
It may happen that the DRS method decides all summation weights to be zero, especially if ϵ_{DD} is chosen too strictly. In this case we need to accept some problematic influence on the pressure sub-problem: we reset that δ_k^i back to 1, for which we expect the least negative impacts, i.e., the respective pressure sub-block shows the smallest lack of diagonal dominance for this block i. Then we can continue as above and obtain a non-singular scaling operator.

4.4.3 Summary and Results

The DRS transformation excludes potentially problematic rows from a total pressure description. This aims at avoiding difficulties for AMG in constructing a hierarchy of levels. At the same time, we can expect the total pressure description to be beneficial for the System-AMG convergence. Even with excluding some rows,

we are still approximating such a description[5]. With this combination of using a total pressure description and at the same time shielding AMG from probably problematic influences, we are able to solve all problems under consideration.

Algorithm 4.1 DRS to determine block $[C_L]_i$ of size $b \times b$

$[C_L]_i \leftarrow \mathbb{1}$
for $j = 1, ..., b$ **do**
  if $[a_{j,1}]_{i,i} < 0$ **then**
    $\delta_j^i \leftarrow 0$
  else
    if $\frac{|[a_{j,1}]_{i,i}|}{\sum_{k \neq i} |[a_{j,1}]_{i,k}|} < \epsilon_{DD}$ **then**
      $\delta_j^i \leftarrow 0$
    else
      $\delta_j^i \leftarrow 1$
if $\delta_1^i \neq 0$ **then**
  for $j = 2, ..., b$ **do**
    if $\frac{\sum_{k \neq i} |[a_{1,j}]_{i,k}|}{|[a_{1,1}]_{i,i}|} < \epsilon_{ps}$ **then**
      $\delta_j^i \leftarrow 0$
else
  if $\forall j = 1, ..., b : \delta_j^i = 0$ **then**
    For 'least problematic' l : $\delta_l^i = 1$
  Find l such that $\delta_l^i \neq 0$
  $([C_L]_i)_{l1} \leftarrow 1$
  $([C_L]_i)_{ll} \leftarrow 0$
for $j = 1, ..., b$ **do**
  $([C_L]_i)_{1j} \leftarrow \delta_j^i$

Our final **System-AMG approach for Black-Oil problems** is described as follows:

- We transform the original linear problem with the DRS method. Algorithm 4.1 summarizes how to obtain the i-th block of C_L for a Black-Oil system.
- We apply our basic System-AMG approach from Figure 3.3 to the transformed problem.

In practice, the choices $\epsilon_{DD} = 0.2$ and $\epsilon_{ps} = 0.02$ turned out to give good results. We could robustly and efficiently solve all considered test problems with these settings. We use these choices as default in our System-AMG approach.

[5]It could be an aspect of further research to involve the saturations in the summation weights, i.e., the weights are not only 0 and 1 then. This way, the resulting pressure system would be even closer to a total pressure problem. We do not consider this here, as we do not have the saturations available at the algebraic level. They would need to be supplied by the simulators.

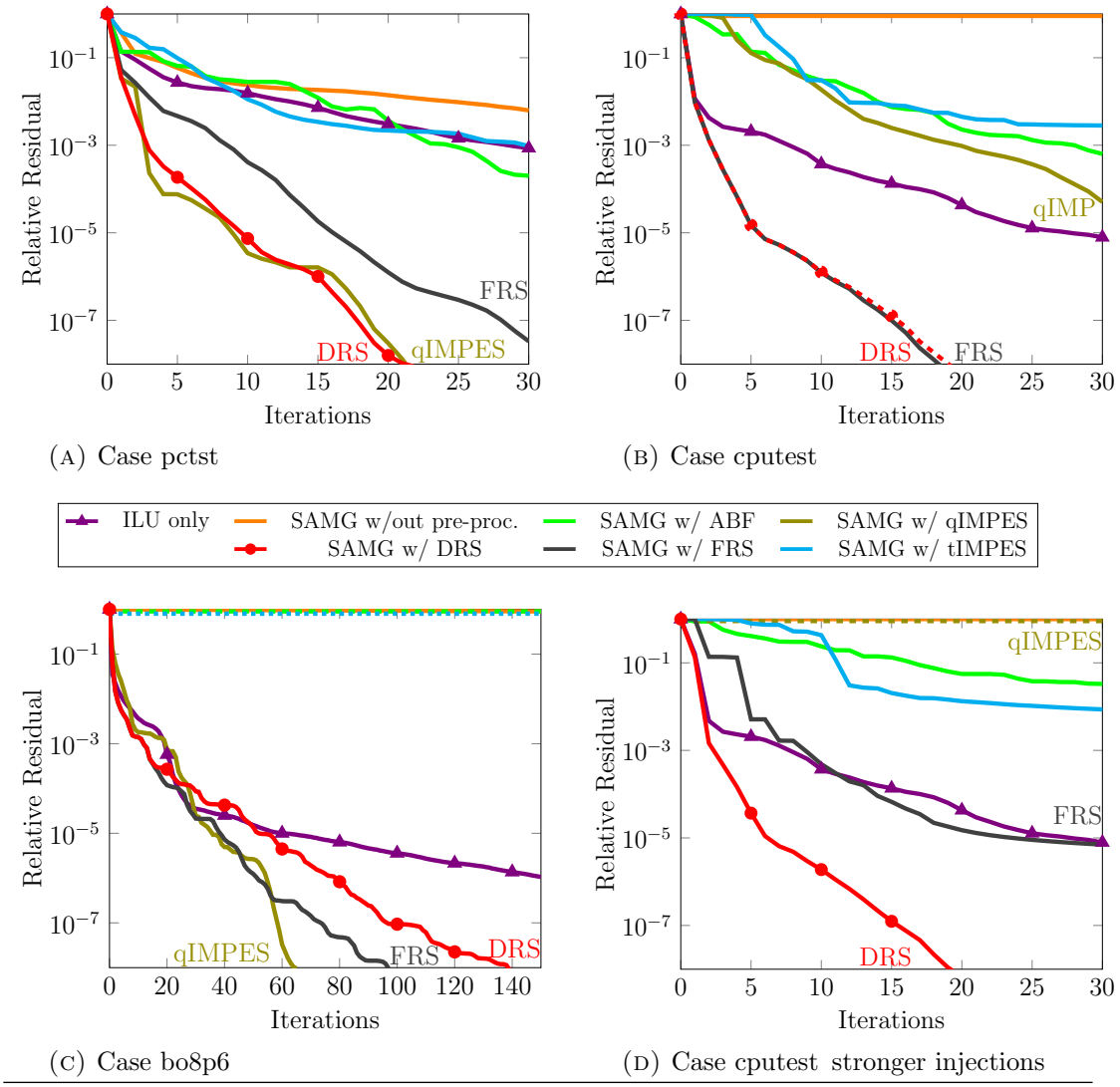
In principal it may happen that all pressure related blocks in the same block-row i suffer from problematic properties. Then, to ensure non-singularity of $\tilde{A}_{pp}$, we have to accept these properties and cannot guarantee that AMG is applicable robustly. While such a situation did not occur in any of our tests, we still could handle it: Either by adjustments at the AMG, or at the simulation side. AMG-adjustments include different choices of the DRS control parameters, or remedy strategies, such as the ones from Remark 4.3.
More promising in such extreme situations, however, are adjustments in the simulator. In Theorem 2.25 we have seen that we can control the diagonal dominance of the pressure related sub-blocks with the time step size. The smaller the time steps, the less critical the pressure sub-blocks. That is, we are guaranteed that our System-AMG approach works properly again, if we decrease the time step size.

In summary, we can regard the DRS method as shielding AMG from problematic influences on an algebraic level as good as possible with a given linear problem. At the same time we attempt to compute a total pressure correction rather than a pressure correction that only reflects a single phase. Because both exploits all pressure related information from the full matrix, we cannot expect more to be possible on a reliable basis. In fact, with the DRS transformation it was possible to efficiently solve a wide range of industrial test cases with System-AMG. This even worked for cases where no AMG-based method was working before.
Figure 4.4 exemplarily demonstrates this for linear systems from selected simulations. The System-AMG approach with the DRS scaling provides a robust and efficient solver method for all four cases. It is the only method that always converges much faster than the one-level reference benchmark (FGMRes/ILU(0)). It is not necessarily the only approach that works efficiently with a given problem. However, for all other approaches we find counter examples where the residual reduction is inefficient compared to the reference method, or there is no convergence at all.
To show the robustness of the DRS method w.r.t. impacts from source terms, in Figure 4.4d we have algebraically strengthened the effect of the existing source terms. The negative impacts from source terms have been scaled[6] by 1.5.

The case bo8p6 is an example for the fact that our choices for ϵ_{DD} and ϵ_{ps} do not

[6]With the sum of the off-diagonal entries, the diagonals in the case of no wells could be approximated.

necessarily need to be optimal for each test case. Clearly, by adjusting these parameters, we can at least achieve the convergence speed of System-AMG after the application of the FRS method. However, these parameter choices have impacts on the convergence with other cases. Our top priority is the robust applicability of the System-AMG solver, which always is much more efficient than a classical one-level method. Hence, we leave it with the above default choices of the control parameters.



(A) Case pctst

(B) Case cputest

(C) Case bo8p6

(D) Case cputest stronger injections

FIGURE 4.4: Convergence history of System-AMG with different pre-processing methods compared to dynamic rowsumming (DRS). DRS is used with the fixed control parameters that worked for all problems under consideration. FGMRes/ILU(0) is given as reference benchmark.
Four representative systems from three dimensional, compressible Black-Oil simulations (cf. Appendix A). (A) and (B) are from Figure 4.1 width 195193 and 29540 cells, respectively. (C) shows bo8p6 with 2.6 million cells and (D) shows cputest (29540 cells) with the effect of injection wells being slightly strengthened algebraically.

Chapter 5

System-AMG for More Complex Reservoir Simulations

The Black-Oil model forms the basis for compositional, thermal and geomechanical models (cf. Figure 2.10). Analogously, our linear solution approach for Black-Oil models serves as a basis for these more complex applications. In this chapter we will extent the System-AMG approach from Section 4.4.3 to problems with additional physical unknowns. This will also exploit the ability of System-AMG to apply a hierarchy to different physical unknowns at the same time. Our final approach will depend on the concrete simulation model, as they lead to linear systems with different properties. However, as long as the model and the involved physical unknowns are made known to the solver, our System-AMG approach serves as a black-box solver for the types of models that we are considering in the scope of this thesis.

5.1 Compositional Simulations

In compositional simulations we have to distinguish the two different modeling approaches, *Natural Variables* and *Volume Balance*. In Section 2.6 we have seen that both approaches lead to different, yet related, linear systems. In our System-AMG approach, we have to take these differences into account.

5.1.1 Natural Variable Formulation

In Natural Variable Formulations we have seen that the unknowns can be divided in primary and secondary ones. The Jacobian (2.38) consists of four blocks:

$$J_{NVF} = \begin{pmatrix} A_{pr,pr} & A_{pr,sc} \\ A_{sc,pr} & A_{sc,sc} \end{pmatrix}. \tag{5.1}$$

In Section 2.6.1 we have stated the following properties for the four different sub-matrices:

- $A_{pr,pr}$ has the same properties as a system from a Black-Oil simulation has, just its dimension has increased.
- $A_{pr,sc}$ has the same properties as A_{ss}.
- The sub-blocks of $A_{sc,pr}$ are diagonal with non-positive entries.
- The sub-blocks of $A_{sc,sc}$ are diagonal and non-negative with $diag(A_{sc,sc}) > 0$.

In the following we are going to discuss how to deal with the different types of unknowns with System-AMG.

5.1.1.1 Handling Primary Unknowns

In Corollary 2.35 we have seen that $A_{pr,pr}$ has the same properties as linear systems from Black-Oil systems. This implies that we have a pressure sub-block A_{pp} with the same properties as we had so far: under mild conditions it is an M-matrix, however, wells may result in indefiniteness. The same analogously holds for the A_{sp} part of $A_{pr,pr}$. The only difference compared to the Black-Oil case is that it now does not only contain two sub-blocks (i.e., is a 2×1 block-matrix), but the number of involved sub-blocks depends on the number of considered components. The blocks still result from pressure derivatives of mass balance equations. That is, they have the same background and comparable properties.
The saturation sub-block A_{ss} now corresponds to saturations and concentrations. This involves more unknowns than in the Black-Oil case. However, each sub-block still is a Z-matrix and A_{ss} is (block) diagonally dominant. Finally, for A_{ps} we have the same properties as we had in the Black-Oil case.
Due to this nearly identical properties of the linear systems, our System-AMG approach, including the DRS transformation, is directly applicable for the linear system that is described by $A_{pr,pr}$.

We just introduce one modification for the DRS method. Due to the possibly high number of components, the dimension of $[C_L]_i$ may be much higher than three. Hence, the DRS method combines a large number of rows in the summation process. Due to the dynamic weights, the concrete number of rows may vary drastically for different blocks. This may result in an unnatural scaling of the different rows of the resulting pressure sub-problem. Therefore, we prefer to scale all summation weights for a block $[C_L]_i$ by a $\kappa_i > 0$, such that

$$\kappa_i \sum\nolimits_j \delta_j^i = 1. \tag{5.2}$$

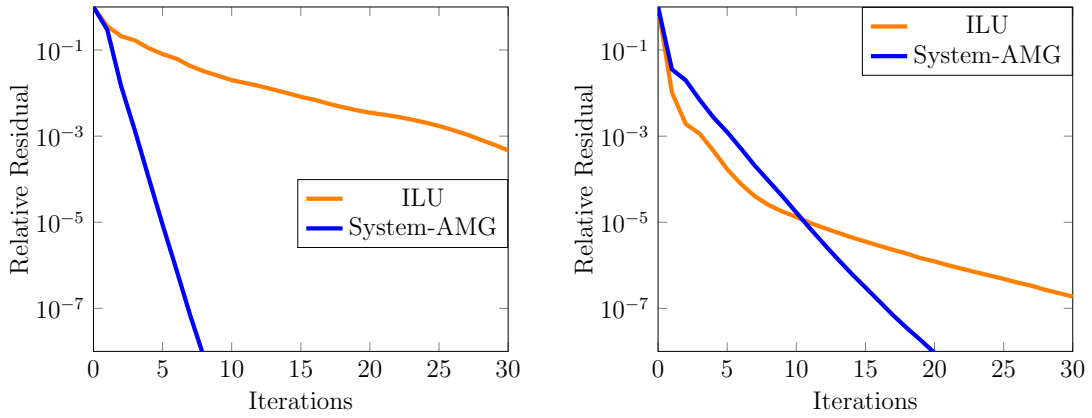
In the Black-Oil case with only three unknowns, this scaling turned out to only have minor effects.

5.1.1.2 Solving for Secondary Unknowns

In the point-wise numbering $A_{sc,sc}$ is block-diagonal. Hence, we expect any iterative method to solve for these unknowns sufficiently well. Moreover, the coupling between primary unknowns and the secondary ones, $A_{pr,sc}$, is of the same structure as the coupling between pressure and the other primary unknowns, A_{ps}. Hence, to design a first System-AMG approach, we can adjust our approach from the Black-Oil case as follows:

- We treat all saturations and concentrations, primary and secondary ones, as we treated saturations in the Black-Oil case: they are forced to remain at the finest level.
- The DRS weights for the secondary unknowns are set to zero. The respective matrix parts do not result from mass balance equations, but have a completely different background than A_{sp} and A_{pp}.

Figure 5.1 shows that this approach is an efficient solver. However, in the following we are going to see that it generally is advisable to treat the secondary unknowns differently.



(A) Case01 from GPRS with 9 Components

(B) SPE10 with 4 Components

FIGURE 5.1: Convergence history of the System-AMG approach for representative systems from NVF compositional simulations. The discretization grids consist of 7500 and 1.1 million cells, respectively (cf. Appendix A).

5.1.1.3 Decoupling Secondary Unknowns

The relations that describe the secondary unknowns are of a completely different background than the ones from primary unknowns. They result from equations that describe flash calculations, and not mass balances. This implies that the scale between both types of matrix parts in general might differ to a certain extent. This did not seem to be an issue for our first System-AMG approach of the previous section and the test cases we have considered. However, a different scale of the respective matrix parts may well give rise to difficulties. We will observe such issues in the context of geomechanical simulations in Section 5.3.

We can avoid such issues to begin with. This is achieved by decoupling the secondary unknowns from the system:

$$\begin{pmatrix} \mathbb{1} & -A_{pr,sc}A_{sc,sc}^{-1} \\ & \mathbb{1} \end{pmatrix} \begin{pmatrix} A_{pr,pr} & A_{pr,sc} \\ A_{sc,pr} & A_{sc,sc} \end{pmatrix} = \begin{pmatrix} A_{pr,pr} - A_{pr,sc}A_{sc,sc}^{-1}A_{sc,pr} & 0 \\ A_{sc,pr} & A_{sc,sc} \end{pmatrix}. \tag{5.3}$$

Obtaining this Schur complement is computationally inexpensive, as $A_{sc,pr}$ and $A_{sc,sc}$ consist of diagonal blocks.

This decoupling is also favorable for another reason: it reduces computational work. In fact, Cao [18] suggested it for this purpose. After the Schur complement is computed, it is sufficient to solve for the primary unknowns. This cuts the degrees of freedom in the reduced system by a factor of two. The secondary unknowns

can be updated afterwards, which, due to the block-diagonality of $A_{sc,sc}$, is rather inexpensive.

However, we still need to address two issues:

- There might be impacts on the non-zero pattern of the matrix by the application of the above Schur complement. Block-wise, we are guaranteed to maintain the pattern of $A_{pr,pr}$. That is, any block $[A_{pr,pr}]_{ij}$ that is zero, remains zero after the decoupling. This holds for two reasons:
  - The non-zero block-pattern of $A_{pr,sc}$ is a subset of the one of $A_{pr,pr}$. This is because the pressure is a primary variable and the pressure blocks' non-zero pattern is not subject to any effects of upstream-weighting.
  - The sub-blocks of $A_{sc,sc}^{-1}A_{sc,pr}$ are diagonal.

  Hence, the non-zero block-pattern of $A_{pr,sc}A_{sc,sc}^{-1}A_{sc,pr}$ is a subset of the one of $A_{pr,pr}$. However, within blocks $[A_{pr,pr}]_{kl}$ that are not zero in $A_{pr,pr}$, the Schur complement might cause changes in the non-zero pattern.
- We need to discuss the applicability of AMG. The above decoupling gives a new sub-block $\tilde{A}_{pr,pr}$ for the primary unknowns. Hence, we need to check whether $\tilde{A}_{pp}$ and $\tilde{A}_{sp}$ still are suited for our System-AMG approach. In fact, in Section 4.2.3 we have seen that a Schur complement decoupling, or an approximation of it, can have serious impacts on AMG's applicability. However, this is different for the particular matrices regarding primary and secondary unknowns. The sub-blocks of $A_{sc,sc}^{-1}$ and $A_{sc,pr}$ are diagonal, and $A_{pr,sc}$ is strongly (block) diagonally dominant. Hence, the Schur complement here mainly influences the diagonals of the sub-blocks in $A_{pr,pr}$. It does so with the 'correct' sign, as we will see in the lemma below. In contrast to this, the approximate Schur complement that we used to weaken the pressure-saturation coupling involved A_{sp}, which typically is only weakly diagonally dominant.

  Lemma 5.1. *Assume a time step Δt that fulfills the requirements from Theorems 2.25 and 2.28 and assume TPFA is used*[1] *(i.e., in the primary block $A_{pr,pr}$ we have A_{pp} being an M matrix, and the blocks in $A_{pr,sc}$ are strongly diagonally dominant M-matrices). Then the pressure sub-block $\tilde{A}_{pp}$ of the primary block $\tilde{A}_{pr,pr}$, after decoupling the secondary unknowns, is an M-matrix.*

[1] With MPFA schemes we could formulate an analog lemma with essentially positive type instead of M-matrices.

Proof. We know that $A_{sc,sc}^{-1}A_{sc,pr}$ consists of diagonal sub-blocks with non-positive entries (cf. Lemmas 2.37 and 2.38, or our repetition at the beginning of Section 5.1.1.2). Following Corollary 2.36, we have the sub-blocks of $A_{pr,sc}$ being Z-matrices and the full matrix is strongly block diagonally dominant. Therefore, the combination of both, $-A_{pr,sc}A_{sc,sc}^{-1}A_{sc,pr}$, consists of blocks that are irreducibly diagonally dominant Z-matrices. By adding this to $A_{pr,pr}$, the new $\tilde{A}_{pp}$ remains an M-matrix. □

Corollary 5.2. *The same holds with* $\tilde{A}_{sp}$. △

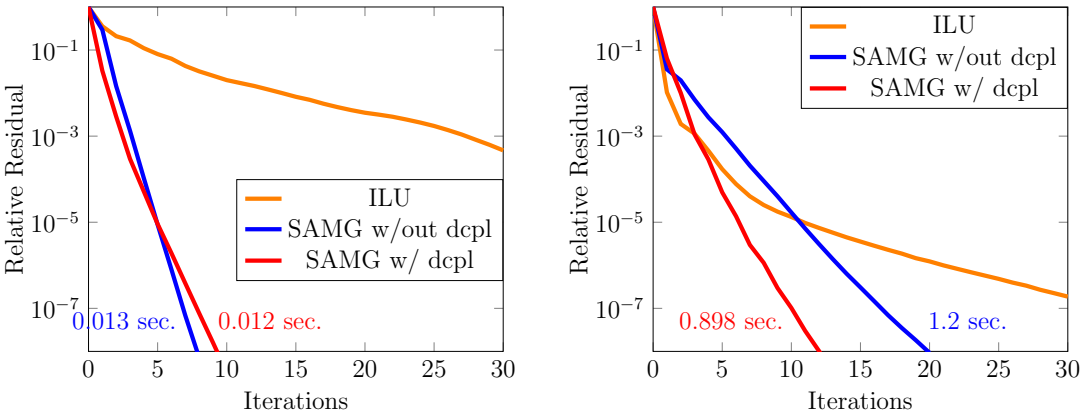
Remark 5.3. *In our discussion on matrix properties we have used Wilson's relation for flash calculations (cf. Section 2.6.1). Then, according to Lemmas 2.37 and 2.38,* $A_{sc,sc}^{-1}A_{sc,pr}$ *essentially is a block-wise scaling by the inverse of the pressure. That is, it is relatively small and we do not expect drastic changes in* $A_{pr,pr}$*. This coincides with practical experience, also with different approaches regarding the flash calculations.* △

5.1.1.4 System-AMG Approach and Results

We can extent our System-AMG approach from Black-Oil simulations to NVF compositional simulations as follows:

- Apply the decoupling of secondary unknowns, as in (5.3).
- DRS transformation for the system that describes the primary unknowns. The DRS is adjusted according to (5.2).
- Use the System-AMG approach from Black-Oil simulations to solve for the primary unknowns. Concentrations are treated like saturations, i.e., they remain on the finest level.
- Explicitly update the secondary unknowns.

Compared to the first System-AMG approach that we used in Figure 5.1, each iteration becomes cheaper. This is at the expense of the Schur complement but typically pays off. Figure 5.2 shows that especially with problems of greater size we can gain significant run time benefits.



(A) Case01 from GPRS with 9 Components

(B) SPE10 with 4 Components

FIGURE 5.2: Convergence history of the System-AMG approach, with and without decoupling of secondary unknowns. The problems result from grids with 7500 and 1.1 million cells, respectively (cf. Appendix A). The run time per iteration is given to show the effect of reducing the degrees of freedom.

5.1.2 Volume Balance Formulation

The situation is more involved in the context of a volume balance formulation approach from Section 2.6.2. Here we have a Jacobian (2.42)

$$J_{VBF} = \begin{pmatrix} A_{VB,p} & A_{VB,N} \\ A_{MB,p} & A_{MB,N} \end{pmatrix}, \tag{5.4}$$

where the pressure unknown is associated with a volume consistency equation. We have the following properties for the four matrix blocks:

- $A_{VB,p} \leq 0$ is a diagonal matrix
- The sub-blocks of $A_{VB,N}$ are diagonal and positive.
- The sub-blocks of $A_{MB,p}$ result from pressure derivatives of mass balances, i.e., $A_{MB,p}$ has the same properties as A_{sp}.
- The sub-blocks of $A_{MB,N}$ result from molar derivatives of mass balances, i.e., $A_{MB,N}$ has the same properties as A_{ss}.

5.1.2.1 Association with Diffusion

The pressure sub-problem $A_{VB,p}$ is a diagonal matrix and we obviously cannot construct an AMG hierarchy based on it that reasonably reflects the fluid flux.

In order to construct a reasonable AMG hierarchy that reflects the pressure differences that induce fluid flux, we need to change the association of unknowns and matrix rows. We could simply exchange rows to associate the pressure with diffusion-based matrix blocks, which results in the system:

$$\begin{pmatrix} A_{MB,p} & A_{MB,N} \\ A_{VB,p} & A_{VB,N} \end{pmatrix} \begin{pmatrix} p \\ N \end{pmatrix} = \begin{pmatrix} f_N \\ f_p \end{pmatrix}. \tag{5.5}$$

As $A_{MB,p}$ has the same properties that we had for A_{sp} in the Black-Oil case, we could directly apply the DRS transformation to the problem and set the last summation weight to zero, as $A_{VB,p}$ is of a different background (due to $A_{VB,p} \leq 0$, it would not be considered by the DRS method anyway). We can expect the result of the DRS to be well-suited to construct an AMG hierarchy.
In fact, this is true. However, setting the last summation weight to zero results in the volume balance being excluded from the system. Hence, the AMG hierarchy would not represent the correct conditions for the fluid flux: It does not reflect volume balance. As a consequence, the pressure correction could allow for more than or less than hundred percent of the pore volume to be filled with the fluid. The post smoothing would have to adjust the pressure correction from the hierarchical process accordingly, which in practice results in significant changes being necessary for the pressure. As a consequence, using our standard System-AMG approach, after only exchanging rows, generally is not faster than using a one-level method only.

Remark 5.4. *For the natural variable formulation this was not a problem, because we directly express the mass balance equations in terms of relative saturations and concentrations, rather than explicit moles. These relative terms, by requiring them to sum to one, imply a volume consistency condition, which is not at all affiliated with the pressure unknown.* △

5.1.2.2 Incorporating the Volume Balance

While, after the exchange of rows, we have the pressure unknown associated with a diffusion based matrix block, this does not take the volume balance condition into account, which now is associated with one of the molar unknowns.
However, our linear system (5.5) has the same properties as the systems (5.1) from NVF simulations. There (cf. Section 5.1.1.3) we could decouple primary and

secondary unknowns in order to solve for the primary unknowns independently of the secondary ones. Transferred to our current situation, we could decouple the molar unknown that now is associated with the volume balance, in order to solve for all other unknowns independently of this unknown. This way, the description of all other unknowns does not only become independent of the respective molar unknown, but also of the volume balance condition that it is associated with.
We split the numbers of moles into two different sets: the last number of moles and all other ones[2]. That is, in $A_{VB,N}$ we have two different blocks: $A_{VB,N} := \begin{pmatrix} A^{\mathcal{A}}_{VB,N} & A^{\mathcal{B}}_{VB,N} \end{pmatrix}$. $A^{\mathcal{B}}_{VB,N}$ refers to the last number of moles and is a square, diagonal matrix. $A^{\mathcal{A}}_{VB,N}$ refers to all other moles and consists of diagonal matrices. We analogously distinguish $A_{MB,N} := \begin{pmatrix} A^{\mathcal{A}}_{MB,N} & A^{\mathcal{B}}_{MB,N} \end{pmatrix}$. Then, for the original system (5.4), we can define the scaling

$$\begin{pmatrix} -A_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1} & \mathbb{1} \\ \mathbb{1} & \end{pmatrix} \begin{pmatrix} A_{VB,p} & A^{\mathcal{A}}_{VB,N} & A^{\mathcal{B}}_{VB,N} \\ A_{MB,p} & A^{\mathcal{A}}_{MB,N} & A^{\mathcal{B}}_{MB,N} \end{pmatrix}$$
$$= \begin{pmatrix} A_{MB,p} - A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A_{VB,p} & A^{\mathcal{A}}_{MB,N} - A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A^{\mathcal{A}}_{VB,N} & 0 \\ A_{VB,p} & A^{\mathcal{A}}_{VB,N} & A^{\mathcal{B}}_{VB,N} \end{pmatrix}. \tag{5.6}$$

This scaling combines two steps:

- It exchanges the rows such that the pressure is associated with a diffusion based matrix block, just as in the previous section.
- It then decouples the one molar unknown, which we can see as a secondary unknown in the NVF-sense, and this way decouples the volume balance.

Regarding the applicability of AMG, we can use the same argumentation as in the NVF-case:

Lemma 5.5. *Assume a time step Δt that fulfills the requirements from Theorems 2.25 and 2.28 and we used TPFA schemes[3] (i.e., the sub-blocks of $A_{MB,p}$ are M-matrices and the mole related blocks in $A_{MB,N}$ are diagonal dominant ones). Then the sub-blocks of $A_{MB,p} - A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A_{VB,p}$ are M-matrices.*

Proof. Under the assumed restriction on Δt, according to Corollary 2.42, the sub-blocks of $A_{MB,N}$ are diagonally dominant M-matrices. $A_{VB,p} \leq 0$ is diagonal and

[2]Distinguishing the last number of moles is completely arbitrary. Distinguishing any other number of moles was sufficient, but made the notation more complex.

[3]With MPFA schemes we could formulate an analog lemma with essentially positive type instead of M-matrices.

the sub-blocks of $A_{VB,N}$ are diagonal with non-negative entries. Hence, we can directly apply our argumentation from Lemma 5.1. □

Corollary 5.6. *With the same time step restriction as in the above lemma,* $A^{\mathcal{A}}_{MB,N} - A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A^{\mathcal{A}}_{VB,N}$ *is strongly (block) diagonally dominant and its sub-matrices are M-matrices.* △

According to the above lemma and corollary, in the remaining linear system we have the same properties as in the primary sub-system with the Natural Variable approach. Hence, the System-AMG approach from Section 5.1.1.1 is applicable. That is, we apply the DRS matrix transformation and construct the AMG hierarchy based on the resulting pressure sub-problem. The hierarchy is applied to the pressure unknown, only, and the numbers of moles remain on the finest level.

We need to discuss the possibility of additional fill-in for the non-zero structure. Completely analogous to our discussion in the case of Natural Variable Formulations, we can conclude this can only affect blocks $[J]_{ij}$ that already involved non-zero terms. However, within such blocks, there generally is a change in the non-zero pattern.

5.1.2.3 Minimize Memory Requirements

We can use a different decoupling to ensure that there is no additional fill-in for the system. We have stressed that we need to decouple the volume consistency because we need to take the volume constraints into account in the AMG hierarchy. That is, we exploit the influences of the decoupling on the pressure sub-problem, rather than the decoupling itself. Therefore, it is sufficient to only influence the pressure related equations by the decoupling. There is no need to also apply changes to the molar related matrix parts. This can be achieved by combining a left and a right scaling:

$$\begin{pmatrix} & \mathbb{1} \\ \mathbb{1} & \end{pmatrix} \begin{pmatrix} A_{VB,p} & A^{\mathcal{A}}_{VB,N} & A^{\mathcal{B}}_{VB,N} \\ A_{MB,p} & A^{\mathcal{A}}_{MB,N} & A^{\mathcal{B}}_{MB,N} \end{pmatrix} \begin{pmatrix} \mathbb{1} & & \\ & \mathbb{1} & \\ -(A^{\mathcal{B}}_{VB,N})^{-1}A_{VB,p} & & \mathbb{1} \end{pmatrix} \tag{5.7}$$
$$= \begin{pmatrix} A_{MB,p} - A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A_{VB,p} & A^{\mathcal{A}}_{MB,N} & A^{\mathcal{B}}_{MB,N} \\ 0 & A^{\mathcal{A}}_{VB,N} & A^{\mathcal{B}}_{VB,N} \end{pmatrix}.$$

The left scaling only exchanges rows, as in (5.5), whereas the right scaling incorporates the volume consistency in the pressure sub-blocks of the matrix. Note that all the four matrices in (5.7), the two scalings and the original and the scaled Jacobian, are square matrices of the same dimension: According to our previous discussion, with a grid of m cells and with c components, there is $A_{VB,p}, A^{\mathcal{B}}_{VB,N} \in \mathbb{R}^{m\times m}$. We have $A_{MB,p}, A^{\mathcal{B}}_{MB,N} \in \mathbb{R}^{cm\times m}$, $A^{\mathcal{A}}_{VB,N} \in \mathbb{R}^{m\times(c-1)m}$ and $A^{\mathcal{A}}_{MB,N} \in \mathbb{R}^{(c-1)m\times(c-1)m}$.

Because all transmissibilities are pressure dependent, $A_{MB,p}$ is non-zero wherever $A^{\mathcal{B}}_{MB,N}(A^{\mathcal{B}}_{VB,N})^{-1}A_{VB,p}$ is. Hence, there is no change in the non-zero pattern of the overall system. Moreover, we compute the same pressure related $\tilde{A}_{MB,N}$ as with the scaling from (5.6). That is, we involve the volume consistency constraint in the construction of the AMG hierarchy. Our result from Lemma 5.5 persists and after the transformation we can apply the same System-AMG approach as before.
We note that $A^{\mathcal{B}}_{VB,N}$ and $A_{VB,p}$ are diagonal. Hence, both scalings, the left and the right one, can be expressed as block-diagonal scalings for the point-wise ordered Jacobian. Their application is rather inexpensive.

Remark 5.7. *In the NVF context, we could of course also use such a right, instead of a left scaling, in order to decouple primary and secondary unknowns. However, because pressure and saturations/concentrations are primary unknowns, it would not have the advantage of avoiding additional fill-in. Due to the upstream-weighting, the non-zero pattern of the saturation/concentration related matrix parts might differ between the phases.* △

5.1.2.4 System-AMG Approach and Results

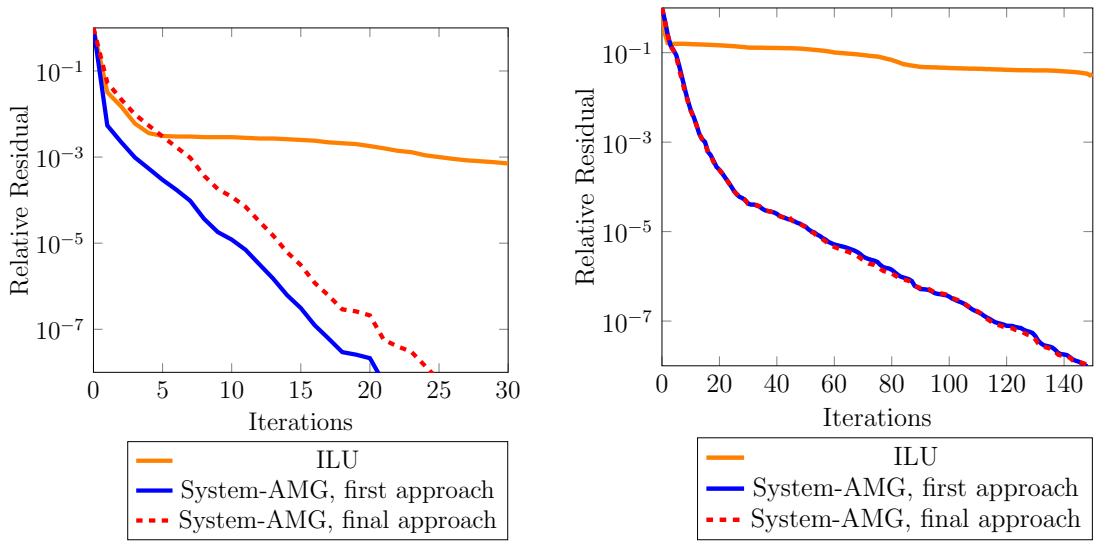
We extent our System-AMG approach from Black-Oil simulations to VBF compositional simulations:

- Apply the decoupling as in (5.7). Compared to the one from (5.6) it has the advantage of not requiring changes in the non-zero pattern of the Jacobian.
- Apply a DRS transformation that excludes the last molar unknown, which now is affiliated with the volume balance equation and no longer with a mass balance. However, after the decoupling, the respective pressure sub-block is zero anyway.

  As the simulation might consider a high number of components, just as in the NVF case, the DRS is adjusted according to (5.2).

- Use the System-AMG approach from Black-Oil simulations to solve for the pressure and the numbers of moles.

Figure 5.3 compares this System-AMG approach with the previous one (i.e., with the decoupling according to (5.6)) and FGMRes/ILU(0) as a reference benchmark. It shows that indeed both methods have comparable convergence rates. We especially have a much more efficient solution method than the one-level approach.



(A) Case 1 (B) Case 2

FIGURE 5.3: Convergence history of the System-AMG approach for representative systems from VBF compositional simulations.
(A) 8 components, discretized in 94,093 grid cells and (B) 9 components, discretized in 963,536 grid cells (cf. Appendix A).

5.2 Thermal Simulations

For thermal simulation models (cf. Section 2.8) the matrix properties are identical to the ones in the respective isothermal Black-Oil or compositional simulation, except that one additional temperature unknown is involved. Hence, regarding the treatment of the temperature, our System-AMG approach does not differ between the Black-Oil and the compositional situation. We will therefore discuss this treatment for the Black-Oil case only.
However, in the compositional case we still apply the previously discussed decoupling transformations and need to consider their impact on the temperature related matrix parts. We will see that the decoupling still results in a system with basically the same properties as in the Black-Oil case.

5.2.1 Black-Oil Thermal Simulations

In Section 2.8 we have seen that, compared to the isothermal Black-Oil case, the linear system is just extended by a temperature unknown. More precisely, it is described by the following matrix:

$$J_{thermal} = \begin{pmatrix} A_{pp} & A_{ps} & A_{pT} \\ A_{sp} & A_{ss} & A_{sT} \\ A_{Tp} & A_{Ts} & A_{TT} \end{pmatrix}. \tag{5.8}$$

A_{pp}, A_{Tp} and the sub-blocks of A_{sp}, except for the influence of source terms, are M-matrices. A_{TT} results from energy diffusion, convection and accumulation. It is an M-matrix with a diagonal dominance that in each row depends on which of the three origins dominates. The remaining matrix parts are strongly diagonally dominant.

As, compared to the isothermal case, the properties of the saturation (or concentration) related matrix blocks did not change, we still force these unknowns to remain at the finest level. Due to the diagonal dominance of the respective sub-matrices, the fine-level smoother will properly solve for them. That is, only pressure and temperature will be subject of a hierarchical treatment by AMG.
Not only the structure of the saturation (or concentration) matrices did not change, also the structure of the coupling between pressure and saturations (or concentrations) is as in the isothermal case. The coupling between temperature and saturation is of the same structure. The same holds for the counter-couplings (i.e., saturation to pressure and temperature, respectively). That is, we can expect the interplay between a coarse grid correction for the pressure-temperature sub-problem, and the fine-level smoother for all unknowns, to work as well as it did in the isothermal case.

Due to the energy balance involving diffusive effects, the application of System-AMG to the coupled pressure-temperature system should be efficient. This can only be realized by the unknown-wise approach: The fluid's permeability K and the energy's one, K_T, in general are not related at all. The reason is that the rock is able to transport energy, i.e. heat, while fluid is not able to flow through rock, only through pores. These differences in the permeabilities do not allow for an efficient point-wise AMG approach.

By considering the pressure-temperature sub-problem, our unknown-wise System-AMG approach constructs a pressure hierarchy based on A_{pp} and a temperature hierarchy based on A_{TT}. In order to properly construct these hierarchies, both sub-matrices need to meet AMG's requirements:

- In Section 2.8.1 we have seen that the sub-problem A_{TT} results from convective and diffusive processes and the accumulation of energy, where diffusion and accumulation dominate. In Corollary 2.45 we have found A_{TT} to be an M-matrix under rather natural conditions. Hence, A_{TT} generally is well-suited to construct an AMG-hierarchy based on it. Depending on how strong the diffusive influences are in a particular time step, AMG will provide more or less benefits regarding the convergence rate. As A_{TT} also involves terms that result from the accumulation of energy, there might be diagonally dominant rows in A_{TT}. Our AMG coarsening (cf. Figure 3.3) will automatically omit constructing a hierarchy for such matrix parts, as the smoother at the finest level can sufficiently solve for the respective variables of the unknown vector. That is, the adaptivity of AMG's setup phase ensures a hierarchy to be locally constructed only where it is beneficial.
- In the pressure related blocks we are faced with the same difficulties that we have observed for the Black-Oil problems. Hence, we include the DRS transformation in our solution approach for thermal problems and need to decide whether or not this shall incorporate the temperature related row. We will not do so for two reasons:
  - The pressure dependence of the energy balance equation (2.46), in contrast to the mass balances, also involves the total enthalpy. Thus, A_{Tp} may be scaled quite differently compared to the total pressure system that the DRS method computes in isothermal simulations.
  - Including the temperature rows in the summation process would mix A_{TT} and A_{pT}. The properties of both are quite different and there would be unnatural couplings between pressure and temperature, especially also in the off-diagonals.

  However, we can still expect the DRS scaling to be beneficial regarding the overall convergence: The energy balance equation involves the total pressure, which is what our coarse grid process computes a correction for, if the system was transformed by DRS.

5.2.2 Compositional Thermal Simulations

We need to consider the impact of the decoupling transformations that we have introduced for the compositional simulations. We will see that we can still apply the Black-Oil System-AMG approach for thermal simulations for the system after applying the transformations from Section 5.1 (cf. (5.7) and (5.3)).

5.2.2.1 Natural Variable Formulation

The temperature is a primary unknown, because the energy balance involves interactions between different grid cells. As we decouple all secondary unknowns, we need to consider the effect of this decoupling on the temperature related sub-blocks in $\tilde{A}_{pr,pr}$. In our discussion from Section 2.8.4.1, we have seen that the properties of $A_{pr,sc}, A_{sc,pr}$ and $A_{sc,sc}$ do not differ between the thermal and the isothermal case. Regarding the properties of $\tilde{A}_{TT}$, after the decoupling of secondary unknowns, we can state:

Lemma 5.8. *Assume the conditions from Corollary 2.45 to hold (i.e., A_{TT} is an M-matrix). Then $\tilde{A}_{TT}$ in $\tilde{A}_{pr,pr} = A_{pr,pr} - A_{pr,sc}A_{sc,sc}^{-1}A_{sc,pr}$ is an M-matrix.*

Proof. Just as in the isothermal case (cf. Lemma 5.1), $A_{sc,sc}^{-1}A_{sc,pr}$ consists of diagonal sub-matrices with non-positive entries. Regarding the temperature related entries of $A_{pr,sc}$, from Corollary 2.47 the sub-blocks are diagonally dominant M-matrices. Hence, the argumentation from Lemma 5.1 applies to the thermal case as well. □

Consequently, after the decoupling of secondary unknowns, as we only need to solve for the primary unknowns, it is sufficient to apply a System-AMG approach for the linear systems from thermal Black-Oil simulations. We just treat the concentrations analogously to saturations.

5.2.2.2 Volume Balance Formulation

Our final matrix transformation (5.7) does not apply any changes to temperature related sub-blocks of the Jacobian matrix. This decoupling only results in changes in the pressure related matrix column of the Jacobian, i.e. $\tilde{A}_{MB,p}$.

As we moreover do not involve the temperature related column in the decoupling (5.7), the Jacobian after the decoupling is nearly the same as in the isothermal case: It is just extended by the temperature unknown. In the isothermal case of a VBF compositional problem, the decoupling results in a linear system with the same properties that we have in an isothermal Black-Oil case. Hence, a System-AMG approach that is applicable to a thermal Black-Oil system is sufficient for a linear system from a thermal VBF simulation.

5.2.3 System-AMG Approach and Results

Summarizing, our System-AMG approach for thermal simulations is as follows:

- If it is a compositional problem, apply the respective decoupling (depending on whether it is NVF or VBF).
- Apply the DRS transformation, but do not include the temperature row, i.e., set the respective summation weight to zero.
- Use the unknown-wise System-AMG approach. A coarsening is only applied to pressure and temperature. Saturations and concentrations remain at the finest level.
- Use the same smoothing as in the Black-Oil case: C/F-wise Gauss-Seidel within the hierarchy and ILU(0) post-smoothing for all unknowns at the finest level.

All other settings are just as in the Black-Oil case (cf. Figure 3.3).

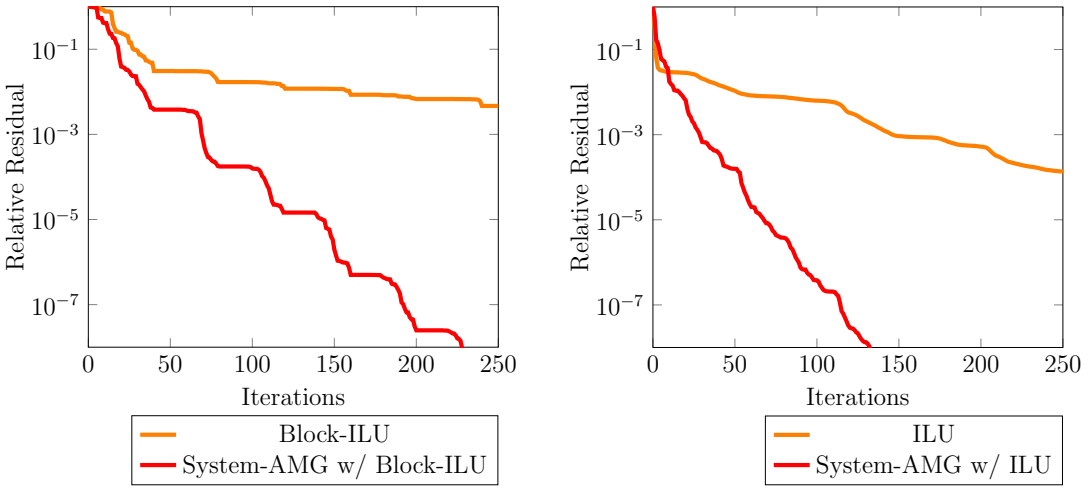
We demonstrate the efficiency of the System-AMG approach for two matrices from an industrial steamflood simulation, i.e., steam is injected instead of water. Both matrices are from the same simulation, just from different time steps.
The first matrix from Figure 5.4a is an example for a linear system where classical incomplete factorization techniques failed[4] to converge. ILU(0) achieved a residual reduction of 0.6 within 1000 iterations. Block-ILU(0) improved this a bit and

[4]A different ordering might allow the ILU to converge. However, as our objective is the application of AMG-based methods, we do not follow this direction. Even if there was an ordering that allowed ILU, or block-ILU, to converge, we can expect the AMG-based method to converge faster.

reduced the residual by a factor of 0.003 within 1000 iterations, but then the residual reduction stagnated: after 10,000 iterations nearly no further reduction was achieved.
This does not necessarily mean that the entire simulation fails. However, the simulator at least has to step back one time step and (drastically) reduce the time step size. Both, of course, is not desired in terms of computational performance.
In fact, this particular linear system also is challenging for System-AMG. However, by choosing block-ILU(0) instead of ILU(0) as a fine-level post-smoother, and by restarting FGMRes after 40 instead of 30 iterations, we obtain eight orders of residual reduction. We have already mentioned that block-ILU(0) is a reasonable option in challenging cases. In fact, both adjustments are only a minor change of the System-AMG approach. They can be activated "on-the-fly", when realizing that the default approach faces serious difficulties. We do neither need to re-compute the AMG hierarchy, nor adjust the time step size.
For reasons of consistency, the adjusted restart of FGMRes also applies to the reference benchmark with block-ILU(0). Note that the 'staircase-like' behavior of the residual results from the GMRes-restarts.
In Figure 5.4b, for the second test case, we compare the convergence history of our default System-AMG approach with FGMRes/ILU(0). This linear system is less challenging and ILU(0) does converge. However, System-AMG reaches the desired accuracy much faster.

5.2.4 Necessity of an AMG Hierarchy for the Temperature

In contrast to the Black-Oil and compositional simulations, in thermal problems we exploit the ability of System-AMG to deal with different physical unknowns in the hierarchy. While, due to the elliptic components in the description of the temperature, this proceeding is rather natural, we have not yet discussed whether this really was necessary. The temperature sub-part A_{TT} does not only result from energy diffusion, but also energy accumulation and convection have an influence. Especially due to the accumulation, at least locally in certain parts of the domain, A_{TT} often is diagonally dominant and here the ILU post-smoother at the finest level might already be a sufficient solver.
However:

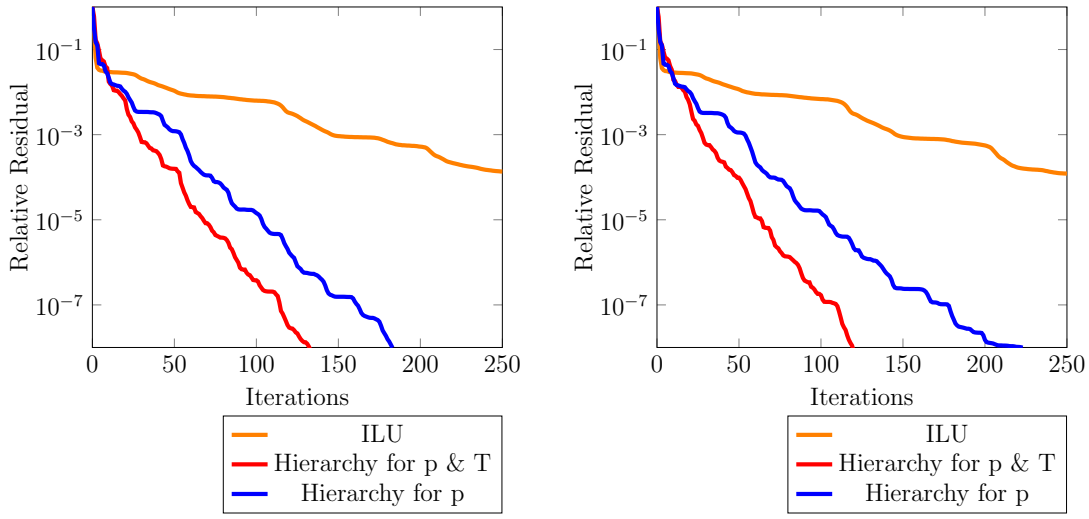


(A) Time step A (B) Time step B

FIGURE 5.4: Convergence history of System-AMG for problems from thermal, two-phase simulations (steam injection). The reservoir is discretized in 235,224 grid cells. The two matrices are from different time steps of the same simulation (cf. Appendix A).
The first problem is highly challenging and required Block-ILU as a fine-level post-smoother.

- The coarsening in our AMG approach does automatically not construct a hierarchy where, due to diagonal dominance, it is not needed.
- In parts of the domain where the energy diffusion has a significant influence on the temperature sub-problem, using a hierarchy is beneficial for the overall process.

In Figure 5.5a, for the problem from Figure 5.4b, we compare our System-AMG approach that applies a coarsening to pressure and temperature with a System-AMG approach that applies the coarsening only to the pressure. Both approaches do converge much faster than ILU. However, we see the beneficial effect of considering both pressure and temperature for the AMG hierarchy.
In Figure 5.5b we consider the same matrix with a small algebraic modification: The diagonal dominance in the accumulation-dominated parts of A_{TT} has been reduced by a factor of two in order to demonstrate the effect of a more significant influence of the energy diffusion. In a practical simulation this influence could, for instance, be more significant with larger time steps, as the accumulation's influence is reduced then. In this modified problem, considering pressure and temperature for the AMG hierarchy is even more beneficial.



(A) Time step B

(B) Time step B, modified A_{TT}

FIGURE 5.5: Convergence history of System-AMG for problems from thermal, two-phase simulations (steam injection). The reservoir is discretized in 235,224 grid cells. The first matrix is an original matrix from this simulation. The second is generated from the first with the temperature problem being made less diagonally dominant (i.e., more diffusion dominated).

5.3 Geomechanical Simulations

The situation becomes even more complex in geomechanical simulations. The previous simulation models are extended by the Lamé Equations (2.57) for the mechanical displacement unknowns. It is well-known that unknown-wise System-AMG is an efficient method to solve such linear elasticity problems, as long as a sufficiently large amount of the boundary is described by fixed (Dirichlet) conditions[5]. This is the case in geomechanical simulations, where nearly all boundaries are fixed. Hence, we expect unknown-wise AMG approaches to be applicable for the isolated mechanical sub-problem. This sub-problem is described by the $J_{M,M}$ sub-block of the full Jacobian from a geomechanical simulation:

$$J_{mech} = \begin{pmatrix} J_{F,F} & J_{F,M} \\ J_{M,F} & J_{M,M} \end{pmatrix}. \tag{5.9}$$

The flow related sub-problem $J_{F,F}$ is identical to what we have been concerned with in Black-Oil, compositional or thermal simulations, respectively. For this sub-problem, we can apply the System-AMG approaches that we have developed

[5]If this is not the case, more sophisticated AMG-variants need to be considered that especially properly reflect the rigid body motions (e.g., smoothed aggregation AMG).

so far. We will only consider Black-Oil flow models in the following and outline the situation for more complex flow models afterwards.

As we do not want to solve the problem in a segregated way, we need to apply a coarsening to the pressure and the displacements simultaneously. This can most easily be achieved by the unknown-wise[6] System-AMG approach, as the background of the flow and the mechanical unknowns is quite differently and they do not even need to be discretized on the same grid. The mechanical effects may act beyond the boundaries of the reservoir, i.e., the grid for the mechanical unknowns needs to cover a larger domain.
Clearly, the convergence properties of the unknown-wise System-AMG approach depend on the couplings between the different unknowns. Especially the couplings between flux and mechanics, $J_{F,M}$, depend on the type of mechanical influence that shall be simulated. In the scope of this thesis, we will only consider poroelasticity effects, i.e., the pores of the rock are widened by the fluid pressures. In (2.68) we have seen that the full linear system - with a solution vector x, a right-hand-side f and the time step size Δt - is given as:

$$\begin{pmatrix} \Delta t J_{F,F} & -J_{M,F}^T \\ J_{M,F} & J_{M,M} \end{pmatrix} \begin{pmatrix} x_F \\ x_M \end{pmatrix} = \begin{pmatrix} \Delta t f_F \\ f_M \end{pmatrix}. \tag{5.10}$$

5.3.1 Uzawa-Smoothing

Depending on the concrete application, the sub-problems of fluid flow and mechanics may be scaled quite differently. If this is the case, the smoothing methods that we have used so far might not work properly. A reasonable remedy, especially for systems like (5.10), is known to be provided by Uzawa-type smoothing methods [84]. In the multigrid context, such methods are typically used for linear systems resulting from Stokes equation.
The Uzawa method that we will use, is designed for systems like

$$\begin{pmatrix} A & -B^T \\ B & C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{5.11}$$

[6]It would also be an option to treat flow and mechanical unknowns differently, i.e., in an unknown-wise manner, but to use a point-wise approach for the mechanical unknowns. This, however, is not yet realized in the SAMG software.

with A and C being well-suited for AMG.

A description of Uzawa-type iterations is given by Elman and Golub [38] and the references therein, or, with more attention on the application as an AMG smoother, by Metsch [72]. The Uzawa method (cf. Algorithm 5.2, with a control parameter α_{uzawa}) essentially alternates between both types of unknowns with an inherited Schur complement. This way, different scales and the particular coupling in B are of less severe influence.

As we use the Uzawa method as a smoother, we only approximately solve the systems in both stages, for instance by Gauss-Seidel or ILU. In our System-AMG approach, according to the System-AMG approaches from simulations that did not consider mechanical effects, we choose Gauss-Seidel for the mechanical system and on the coarse levels of the flow system, whereas we use ILU at the finest level of the flow system.

Algorithm 5.2 Uzawa Algorithm for (5.11)

1: **for** $i = 0$ *until convergence, or until* $i > maxIter$ **do**
2: $\quad u_{i+1} = A^{-1}(f + B^T p_i)$
3: $\quad p_{i+1} = p_i + \alpha_{uzawa}(g - Bu_{i+1} - Cp_i)$

5.3.2 Modification of the DRS Transformation

For the flow-related sub-system we still apply the DRS transformation, as we still might be concerned with the same difficulties as before[7].

However, after the DRS transformation for the fluid part, we generally do not have $J_{F,M} = -J_{M,F}^T$ any more. This is a problem for the proper application of Uzawa-type smoothers. However, we can modify the DRS method in order to preserve this property:

$J_{F,M}$ results from the discretized divergence of the displacements. More precisely, from the discretization of the term $\frac{\partial}{\partial t}\rho_\alpha S_\alpha \alpha_{Biot} \nabla u$ in the mass balance (2.63) of phase α under mechanical influences. This applies to all involved phases. Hence, all rows of $J_{F,M}$ that correspond to the same cell i are nearly identical. They only differ by a scaling that depends on the different material properties. We can adjust the non-zero summation weights of DRS according to these scales. This

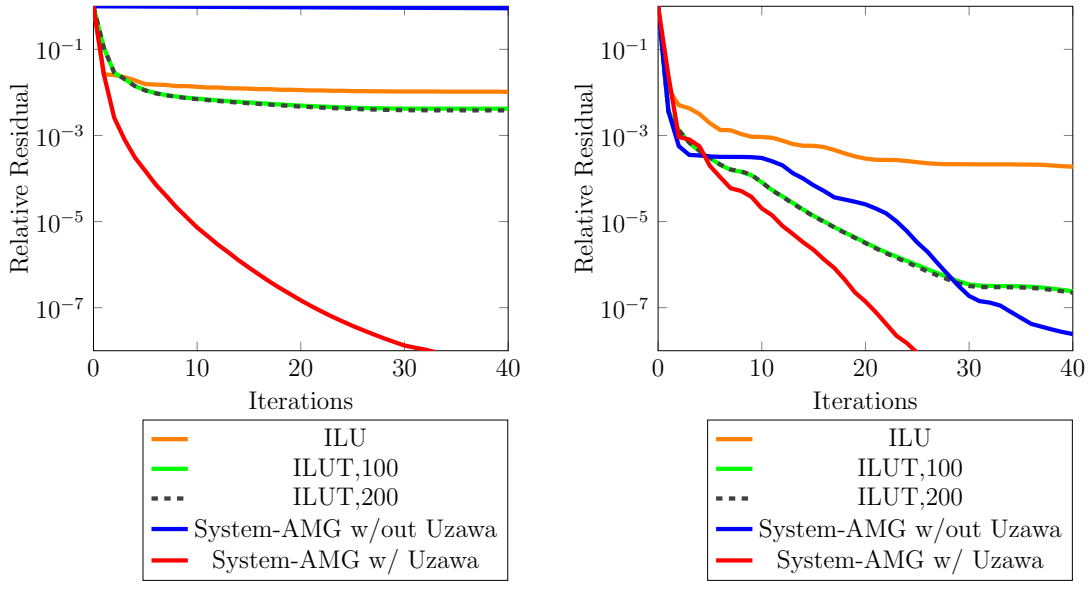
[7] Although, with A_{pp}^{noMech} denoting the pressure problem from a pure fluid flow simulation, we have $A_{pp} = A_{pp}^{noMech} + \frac{1}{\Delta t} P_M$, $\frac{1}{\Delta t}$ and P_M may not be large enough to outweigh the influence of wells.

way, we can ensure that the DRS transformation does not result in any change for $J_{F,M}$, but it still results in a $\tilde{J}_{F,F}$ that is better-suited for System-AMG than the original flow sub-problem might have been.

5.3.3 System-AMG Approach and Results

The linear system (5.10) has the same structure as (5.11) and we can use Uzawa-type smoothing for the linear systems that result from fluid flow coupled to poroelasticity effects. We then do not expect difficulties from differently scaled sub-problems. For a Black-Oil fluid flow model that is coupled to a geomechanical model, our System-AMG approach reads as:

- DRS transformation with the modified weights, as discussed in the previous section
- Unknown-wise System-AMG. Saturations and concentrations remain at the finest level
- Uzawa-type smoothing on each level. We use the Uzawa realization in SAMG, as of version 27z1. α_{uzawa} remains at its default of 0.125.



(A) Cube_Coup_dt0 (B) Cube_Coup_dt6

FIGURE 5.6: Convergence history of the System-AMG approach for representative systems where single phase flow is coupled with geomechanics. Both systems are from the same simulation (3D, 541190 cells), however, at different time step sizes (cf. Appendix A).

Figure 5.6 shows that this approach[8] is a robust solution method. We consider two matrices from the same simulation, but at different time step sizes. This results in differently scaled flow-systems. For the first time step size, System-AMG with Uzawa-type smoothing is the only solver method to be considered, as it can handle the different scalings. For the other time step size, also a straight forward extension of our standard approach (i.e.: unknown-wise coarsening; ILU-smoothing at the finest and C/F Gauss-Seidel smoothing on all other levels) gives an acceptable convergence, although the convergence speed is much faster with Uzawa-type smoothing.
ILU(0) in both cases does not converge in a reasonable number of iterations. We include ILUT [87] in our comparison, as it is widely used as a default solver for geomechanical simulations. We choose a dropping tolerance of 0.005 and a fill-in of 100 and 200 (especially in the mechanical part we typically have to expect large stencil sizes. In the full systems we have roughly 60 non-zeros per row on average). For the problem with the rather harmless scaling differences between the unknowns, we can achieve convergence with ILUT, if we sufficiently increase the fill-in. However, the factorization and each iteration becomes increasingly expensive, our System-AMG approach still converges much faster and for the first matrix ILUT did not succeed.

5.3.4 Compositional and Thermal Flow Problems

In the scope of this thesis we are not concerned with geomechanical models that are coupled to compositional and/or thermal fluid flow models. However, we outline how our System-AMG approach could be extended. In Section 2.9.4.2 we have seen that the properties from (5.10) for the sub-blocks regarding mechanics and flow, and the cross-couplings, persist. Hence, our System-AMG approach with Uzawa smoothing in principle still is applicable. We just need to discuss whether the consideration of further aspects on the flow side, as described in Section 2.9.4.2, has impacts on the overall approach.

- Compositional, Natural Variable Formulation:
  We have mentioned that $J_{F,M}$ is zero in rows that correspond to secondary

[8] As we consider a single phase problem, there are no saturation unknowns to be considered here and, for practical reasons, we can replace ILU at the finest level by Gauss-Seidel.

unknowns. Hence, the decoupling of secondary unknowns from our System-AMG approach for the NVF flow problem does not change $J_{F,M}$. After the decoupling we still have $J_{F,M} = -J_{M,F}^T$ and our Black-Oil System-AMG approach with Uzawa smoothing is applicable.

- Compositional, Volume Balance Formulation:
  Our System-AMG approach involved a left and a right scaling from (5.7). The right scaling of the flow problem $J_{F,F}$ also is applied to $J_{M,F}$. However, as we have seen that all but the pressure related column of $J_{M,F}$ are zero, the particular right scaling from (5.7) does not result in changes for $J_{M,F}$. This is different for the left scaling, which has an impact on $J_{F,M}$. In this thesis we are not concerned with simulations where VBF flow models are coupled with geomechanics and leave this aspect to further research.
- Thermal:
  Our System-AMG approach for thermal simulations does not involve any additional transformation of the linear system. Hence, there is no need for adjustments regarding the geomechanical System-AMG approach.

Chapter 6

Algorithmical Aspects of Performance Improvements

In the previous chapters we have presented an efficient solution approach for solving the linear problems in reservoir simulations by AMG and we have discussed how its robust applicability can be ensured in a wide range of models. In this chapter we are going to discuss two possibilities to improve the computational performance of this solution approach.
First, for a typical linear system from reservoir simulations, we will see that we can exploit the strong material heterogeneities in order to use aggressive coarsening strategies without negative impacts on the convergence speed. These coarsening strategies will turn out to be beneficial in terms of computational performance and memory requirements.
Secondly, we will discuss options to exploit special matrix properties in the case of adaptive implicit simulations. We can reduce the degrees of freedom for our ILU post-smoother, which results in better run times.

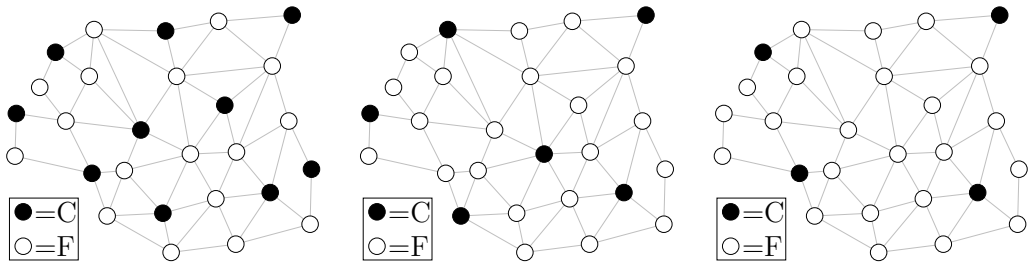
6.1 Aggressive Coarsening in Reservoir Simulation

AMG is not a fixed algorithm, but a methodology to create efficient solvers. That is, we have some freedom in adjusting any AMG approach to a particular application. In this section we will investigate the effects of aggressive coarsening in

reservoir simulations. Initially, aggressive coarsening was designed to reduce the memory requirement of AMG and speed up the AMG cycles, at the expense of a slower convergence. In the context of reservoir simulations, however, such a slow down of the convergence rate is typically not observed and the overall computational time is reduced. We are going to heuristically explain this effect by analyzing a model problem.

6.1.1 Aggressive Coarsening Algorithm

Aggressive coarsening is a variant of the RS coarsening introduced in Section 3.1.1.4. For the classical C/F-splitting, each F-variable is required to be strongly connected to at least one C-variable. Stüben [95], based on Ruge and Stüben [85], extends this idea by the concept of strong *long-range* connectivity. More precisely, $\xi\tau$-aggressive coarsening, w.r.t. the graph of strong connections (3.8), requires each F-variable to be connected to C-variables via at least ξ paths of length at most τ. This extended interpretation of connections results in a smaller set of C-variables than with the original RS coarsening. The set will be the smaller, the smaller ξ and the larger τ. Practically most relevant are the 12 and 22 variants, which are referred to as A1 and A2 aggressive coarsening, respectively. In Figure 6.1 these variants are illustrated and compared with classical coarsening.



(A) RS Coarsening (B) A2 Coarsening (C) A1 Coarsening

FIGURE 6.1: Classical Ruge-Stüben coarsening compared with its A2 and A1 aggressive variants for an exemplary strong adjacency graph. The coarsening ratios are 27:10, 27:6 and 27:4, respectively.

Aggressive coarsening, generally, has three beneficial effects:

- Due to the smaller coarse level operators, AMG's setup phase becomes faster. Also the total number of levels needed to reach a coarsest level problem of reasonably small size may decrease, which again reduces computational work for the setup.

- As the coarse level operators are of smaller dimension, and as smoothing and further transfer operators are applied on smaller (and less) levels, each AMG cycle becomes computationally cheaper.
- The amount of memory needed to store the multigrid hierarchy is significantly reduced.

However, these benefits come at the expense of F-variables not necessarily being strongly connected to a C-variable directly. Hence, in the coarse grid correction process it will be necessary to interpolate some F-variables from other F-variables, once these have been interpolated from strongly connected C-variables themselves. This consecutive interpolation of F-variables is referred to as *multipass interpolation*. As the AMG convergence is highly determined by the "quality" of the interpolation operator, multigrid cycles based on aggressive coarsening typically converge significantly slower than their standard coarsening counter-parts.

In practice, it is common to mix aggressive and classical coarsening. Typically, the first coarse level is constructed using the aggressive approach, while the further levels are created using the classical coarsening. This gives an initially fast reduction of the degrees of freedom, while it limits the negative influence of the multipass interpolation to only one transfer. Such an approach is often used for Scalar-AMG with matrices resulting from non-compact discretizations of diffusion-based problems. This includes Poisson-type equations, but also holds for the pressure problem (2.12) from IMPES simulations. Here, as a matter of experience, the reduced costs per iteration usually outweigh the increased number of iterations. Combined with the faster setup phase, this results in a faster method.

The situation is different with systems resulting from FIM or AIM simulations, the types of problems that we are concerned with in this thesis. Here we are dealing with coupled systems of linear equations, for which the AMG hierarchy is only applied to a part of them. The reduction of computational costs by aggressive coarsening only effects this subset of unknowns. The cost of ILU-smoothing at the finest level, however, is not reduced by the aggressive coarsening. As it involves all unknowns, this smoothing is relatively expensive and it appears questionable whether the benefit per iteration still outweighs the higher number of iterations. Nevertheless, for our System-AMG approach, with an incomplete factorization serving as fine-level smoother, as a matter of practical experience, we can exploit aggressive coarsening for typical problems from reservoir simulations and gain

benefits w.r.t. both memory and performance. This can be motivated by an analysis of ILU smoothing in a model case.

6.1.2 ILU(0) Smoothing Properties in a Model Case

For linear problems resulting from Poisson's equation, with ILU smoothing, just as with relaxation methods, an increase of iterations is observed when switching to aggressive coarsening. However, compared to such completely homogeneous linear problems, the ILU smoother behaves differently if applied to highly heterogeneous reservoir simulation problems. As a motivation, we are going to analyze this for a model problem in a geometric multigrid context. Although much simpler than real simulation problems, the results with this model problem reflect the behavior observed in more complex situations, where a corresponding analysis is practically impossible.
Let us consider the two dimensional anisotropic differential equation

$$-\epsilon\frac{\partial^2}{\partial x^2}u - \frac{\partial^2}{\partial y^2}u = f \text{ with } x, y \in [0,1]^2 \text{ and } u = g \text{ for } x, y \in \partial[0,1]^2, \qquad (6.1)$$

with some $0 < \epsilon \leq 1$, discretized by a 5-point stencil on a structured mesh of resolution h. We assume a lexicographic ordering of the mesh cells. Clearly, the resulting linear system $A_{aniso}u = f_{aniso}$ is described by an spd M-matrix. Hence, it is known (cf. [71]) that the ILU(0) operator S_{ILU} corresponding to any pattern exists and $\rho(S_{ILU}) < 1$.

6.1.2.1 Local Fourier Analysis

Local Fourier Analysis (LFA) is widely used to analyze geometric multigrid approaches. In particular, the smoothing properties of a method like ILU can be quantified, depending on the aggressiveness of coarsening.
The LFA concept goes back to Brandt [16]. For a detailed introduction, and a discussion of its application, we refer to Trottenberg et.al., as well as Wienands and Joppich [97, 107]. We only give a brief excerpt for the 2D case here.
An arbitrary error is assumed to be expanded into Fourier components. The fundamental quantities to describe these components are the grid functions

$$\phi(\Theta, x) = e^{i\Theta\frac{x}{h}}, \qquad (6.2)$$

where $\Theta = (\mu, \nu)$ characterizes the frequency. With the concept of LFA, it can be analyzed how multigrid components act on the individual frequencies. In our discussion, we only consider the smoother. For simplicity, LFA assumes infinite grids, which essentially means that boundary conditions are ignored. For smoothing by relaxation, for instance, this is fairly natural: it is a local process, essentially independent of the boundaries, except from a direct neighborhood of the boundary. Assuming a fixed mesh size h, the Fourier components are identified by $(\mu, \nu) \in [1, 2, ..., \frac{1}{h})^2 =: \Phi \subset \mathbb{N}^2$. With LFA we can determine how each of these components is reduced by the smoothing operator, which yields a *reduction factor* $\lambda_{smoother}(\mu, \nu)$ for each component within one smoothing sweep.
In order to analyze the smoothing properties of an operator, we need to distinguish low- and high-frequency error components, Φ_{low} and Φ_{high}, respectively, with $\Phi_{high} = \Phi \setminus \Phi_{low}$. Which components (μ, ν) are low-frequency, i.e., are in Φ_{low}, depends on the coarsening. Roughly speaking, low-frequencies are those which can still be "represented" properly on the coarser grid. If we assume this grid to be the analogue of the initial grid, just with mesh size $H > h$, the low-frequencies correspond to $\Phi_{low} = [1, 2, ... \frac{1}{H})^2$.
The *smoothing factor* of a given smoother is defined as the worst reduction factor over all high-frequencies. That is:

$$\sigma_{smoother} = sup\{\lambda_{smoother}(\mu, \nu) | (\mu, \nu) \in \Phi_{high}\}. \tag{6.3}$$

The low-frequency components are reduced by the coarse grid correction. Given that the coarse grid correction and the interplay with the smoother works properly, the smoothing factor can be regarded as an approximation of the two-level convergence rate of multigrid. Obviously, the choice of the coarsening, given by the coarse grid's mesh size H, i.e., its aggressiveness, has a direct impact on the smoothing factor. We are going to discuss this in the following.

6.1.2.2 Smoothing Factor of ILU(0)

Wittum [108] discussed the smoothing properties of ILU(0) for the discretization of (6.1), where he observed (cf. Figure 2 in [108]) a convergence peak depending on ϵ and h. He concluded the boundary conditions not to be completely negligible for ILU smoothing and adjusted the LFA accordingly to compute the reduction

factors of S_{ILU}. For ILU(0) they read as:

$$\lambda_{ILU}(\mu,\nu,\epsilon,h) = \frac{\delta|cos((\mu-\nu)h\pi)|}{|1+\epsilon-\epsilon cos(\mu h\pi)-cos(\nu h\pi)+\delta cos((\mu-\nu)h\pi)|}, \tag{6.4}$$

with $0 < \epsilon \leq 1$, $\delta := \frac{\epsilon}{1+\epsilon+\sqrt{2\epsilon}}$ and $(\mu,\nu) \in \Phi$.
By means of (6.4), we can compute the smoothing factor of ILU(0) with classical coarsening. We, however, will use it to compute the smoothing factor with aggressive coarsening. We will moreover consider semicoarsening, as this corresponds to the AMG coarsening for the anisotropic model problem.

According to our considerations from the previous section, we need to find the maximum of $\lambda_{ILU}(\mu,\nu,\epsilon,h)$ over $(\mu,\nu) \in \Phi_{high}$. As a first step, we will see that this maximum can only be found at the boundary $\partial\Phi_{high}$. We follow the argumentation by Wesseling [105] (Example 7.8.1) to show this result:
Instead of maximizing λ_{ILU}, we can equally well minimize $\frac{1}{\lambda_{ILU}}$. A straight forward calculation gives:

$$\frac{1}{|\lambda_{ILU}(\mu,\nu,\epsilon,h)|} = |\frac{1}{\delta}\ \frac{1+\epsilon-\epsilon cos(\mu h\pi)-cos(\nu h\pi)}{cos((\mu-\nu)h\pi)}+1|. \tag{6.5}$$

If we now choose $\phi := \mu - \nu$ and define

$$d = d(\epsilon,h,\mu,\phi) := \frac{1+\epsilon-\epsilon cos(\mu h\pi)-cos((\mu-\phi)h\pi)}{cos(\phi h\pi)}, \tag{6.6}$$

we obtain the minimization problem

$$\frac{1}{\sigma_{ILU}(\epsilon,h)} = min_{(\mu,\nu)\in\Phi_{high}}|\frac{1}{\delta}d(\epsilon,h,\mu,\phi)+1|. \tag{6.7}$$

Lemma 6.1. *Any extremal value of d must be found at $(\mu,\nu) \in \partial\Phi_{high}$.*

Proof. A necessary condition for an extremal value is $\frac{\partial}{\partial\mu}d = 0$, with

$$\frac{\partial}{\partial\mu}d = \frac{(\epsilon sin(\mu h\pi)+sin((\mu-\phi)h\pi))h\pi}{cos(\phi h\pi)}.$$

Hence, an extremal value requires $\epsilon sin(\mu h\pi) = -sin((\mu-\phi)h\pi) = -sin(\nu h\pi)$. But this would be a contradiction to $0 < \mu,\nu < \frac{1}{h}$, as for these values we have $sin(\mu h\pi), sin(\nu h\pi) > 0$. □

In order to find the smoothing factor, we consequently only need to consider (μ, ν) at the boundary $\partial\Phi_{high}$. We will do so for two types of coarsening, and we will visualize our results with an exemplary fine grid resolution of $h = \frac{1}{1024}$.

6.1.2.2.1 Uniform Coarsening

A classical geometric multigrid approach is to coarsen both grid directions analogously. We then have to consider the high-frequency modes in $\Phi_{high} = \Phi \setminus \Phi_{low}$ with $\Phi_{low} = [1, \frac{1}{H})^2$. The concrete choice of H defines the aggressiveness of the coarsening. For the moment we only assume $H > h$. We can identify three outstanding combinations of μ and ν.

Lemma 6.2. *Let any $h, H > h$ and $\epsilon > 0$ be given and define $c_1 := c(h, H, 1, \epsilon)$, with $c(h, H, \omega, \tau)$ from Appendix C. Then the smoothing factor $\sigma_{ILU}(\epsilon, h)$ is the maximum of $\lambda_{ILU}(\mu, \nu, \epsilon, h)$ taken over the following three values of (μ, ν):*

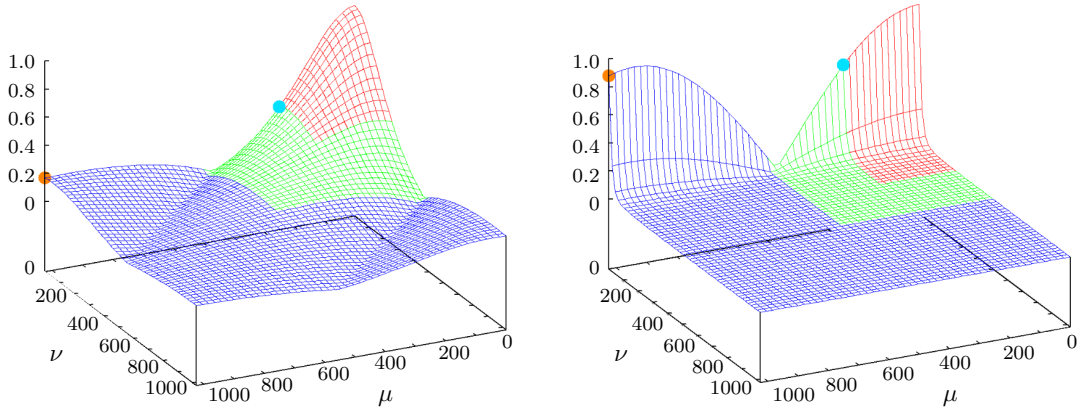
- $(\mu, \nu)_1 := (\frac{1}{h} - 1, 1)$
- $(\mu, \nu)_2 := (\frac{1}{H}, c_1)$
- $(\mu, \nu)_3 := (\frac{1}{H}, 1)$

Proof. See Appendix C.1. □

In the context of geometric multigrid, a classical choice for the coarsening rate is $H = 2h$, while a more aggressive variant, for instance, could be $H = 4h$. With the above result we know that differences in the smoothing factor between both choices can only result from different reduction factors of the presented three components. If we regard them as functions of the anisotropy ϵ, and compute them for an exemplary grid with $h = \frac{1}{1024}$, for small ϵ we find that the smoothing factor is determined by the reduction factor at $(\frac{1}{h} - 1, 1)$, which is independent of H. Table 6.1 shows that this holds for $\epsilon \leq 10^{-\sqrt{2}}$. For such anisotropies, the smoothing factor is identical for both considered coarsening rates. That is, in the case $H = 4h$ we can still expect the same overall convergence rate as in the case $H = 2h$. The situation is illustrated in Figure 6.2, where the reduction factors of the Fourier modes are plotted for different values of ϵ. The values at $(1023, 1)$ and $(256, 1)$ (i.e. $(\frac{1}{h} - 1, 1)$ and $(\frac{1}{4h}, 1)$) are marked.

	$H = 2h$			$H = 4h$		
ϵ	$(\frac{1}{h}-1,1)$	$(\frac{1}{H},c_1)$	$(\frac{1}{H},1)$	$(\frac{1}{h}-1,1)$	$(\frac{1}{H},c_1)$	$(\frac{1}{H},1)$
10^{0}	<u>0.168</u>	0.033	0.035	0.168	<u>0.395</u>	0.361
10^{-1}	<u>0.469</u>	0.068	0.082	0.469	0.440	<u>0.556</u>
$10^{-\sqrt{2}}$	<u>0.596</u>	0.033	0.097	<u>0.596</u>	0.282	0.594
10^{-2}	<u>0.748</u>	0.009	0.113	<u>0.748</u>	0.086	0.626
10^{-3}	<u>0.884</u>	0.001	0.125	<u>0.884</u>	0.007	0.644
10^{-4}	<u>0.887</u>	10^{-4}	0.123	<u>0.887</u>	0.001	0.617
10^{-5}	<u>0.590</u>	10^{-5}	0.081	<u>0.590</u>	10^{-4}	0.405
10^{-6}	<u>0.130</u>	10^{-6}	0.018	<u>0.130</u>	10^{-5}	0.089
10^{-7}	<u>0.015</u>	10^{-7}	0.002	<u>0.015</u>	10^{-7}	0.010
10^{-8}	<u>0.001</u>	10^{-8}	10^{-4}	<u>0.001</u>	10^{-8}	0.001
10^{-9}	<u>10^{-4}</u>	10^{-9}	10^{-6}	<u>10^{-4}</u>	10^{-9}	10^{-4}

TABLE 6.1: Reduction factors at combinations (μ, ν) that are relevant in the case of standard coarsening.



(A) $\epsilon = 1.0$ (B) $\epsilon = 10^{-3}$

FIGURE 6.2: Reduction factors of the ILU(0) smoother with different values of ϵ. With coarsening in both directions, high-frequency modes correspond to the blue area in the case of $H = 2h$ and to the union of the blue and green area in the case of $H = 4h$. The points $(\frac{1}{h}-1,1)$ and $(\frac{1}{4h},1)$ are marked in orange and light blue, respectively (according to the first and last column in Table 6.1). These reduction factors determine the smoothing factor.

For greater values of ϵ, i.e., rather isotropic problems, this is different. Here the smoothing factor increases for aggressive coarsening, as the reduction factor at $(\frac{1}{H}, 1)$ (or $(\frac{1}{H}, c_1)$) is greater than the one at $(\frac{1}{h} - 1, 1)$. Consequently, although we use ILU(0) as a smoother, for an isotropic problem, we have to expect a slower convergence rate of the two-level method.

6.1.2.2.2 Semicoarsening

Our discussion so far assumed that we were using the same coarsening in each grid direction. While this is a reasonable choice for a geometric multigrid approach, it does not properly reflect the situation of algebraic multigrid. In the isotropic case of $\epsilon = 1$, classical AMG coarsening for a two-level method would have a coarsening rate of $H = \sqrt{2}h$. In the anisotropic case with small ϵ, the AMG coarsening would adapt itself to the matrix structure. This is, as soon as $\epsilon < \epsilon_{str}$, with ϵ_{str} the criterion for strong connections from (3.8), the AMG coarsening is performed in only one direction: along the strong couplings. In the context of geometric multigrid this corresponds to semicoarsening, in our case with small ϵ, along the y-direction. The set of low-frequency modes then is $\Phi_{low} = [1, \frac{1}{h}) \times [1, \frac{1}{H})$.

Lemma 6.3. *With this semicoarsening, the smoothing factor of ILU(0) is given by the maximal reduction factor $\lambda(\mu, \nu, \epsilon, h)$ taken over the three values of (μ, ν):*

- $(\mu, \nu)_4 := (\frac{1}{h} - 1, \frac{1}{H})$
- $(\mu, \nu)_5 := (c_2, \frac{1}{H})$
- $(\mu, \nu)_6 := (1, \frac{1}{H})$

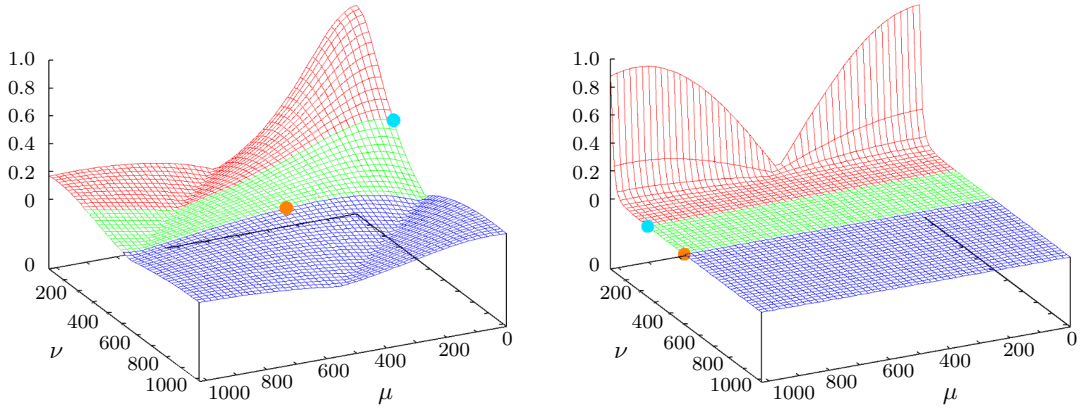
Where, with (C.2), $c_2 := c(h, H, \epsilon, 1)$.

Proof. See Appendix C.2. □

The values of these three different reduction factors, for two different amounts of aggressiveness in the coarsening, are given in Table 6.2, for different choices of ϵ and an exemplary grid of mesh size $h = \frac{1}{1024}$. Also in the case of semicoarsening, the difference in the smoothing factor between both types of aggressiveness decreases the stronger the anisotropy becomes. Especially for $\epsilon \leq 10^{-4}$, there is nearly no difference at all, if we keep in mind the values in the table are rounded. Figure 6.3 visualizes the situation for two choices of ϵ.

	$H = 2h$			$H = 4h$		
ϵ	$(\frac{1}{h}-1, \frac{1}{H})$	$(c_2, \frac{1}{H})$	$(1, \frac{1}{H})$	$(\frac{1}{h}-1, \frac{1}{H})$	$(c_2, \frac{1}{H})$	$(1, \frac{1}{H})$
10^{0}	0.013	<u>0.114</u>	0.031	0.118	0.299	<u>0.36</u>
10^{-1}	0.006	<u>0.045</u>	0.007	0.110	<u>0.132</u>	0.109
10^{-2}	0.001	<u>0.006</u>	0.001	<u>0.020</u>	0.018	0.016
10^{-3}	10^{-4}	<u>0.001</u>	10^{-4}	<u>0.002</u>	0.002	0.002
10^{-4}	10^{-5}	<u>10^{-4}</u>	10^{-5}	<u>10^{-4}</u>	10^{-4}	10^{-4}
10^{-5}	<u>10^{-6}</u>	10^{-6}	10^{-6}	<u>10^{-5}</u>	10^{-5}	10^{-5}
10^{-6}	<u>10^{-7}</u>	10^{-7}	10^{-7}	<u>10^{-6}</u>	10^{-6}	10^{-6}
10^{-7}	<u>10^{-8}</u>	10^{-8}	10^{-8}	<u>10^{-7}</u>	10^{-7}	10^{-7}
10^{-8}	<u>10^{-9}</u>	10^{-9}	10^{-9}	<u>10^{-8}</u>	10^{-8}	10^{-8}
10^{-9}	<u>10^{-10}</u>	10^{-10}	10^{-10}	<u>10^{-9}</u>	10^{-9}	10^{-9}

TABLE 6.2: Reduction factors at combinations (μ, ν) that are relevant in the case of semicoarsening.

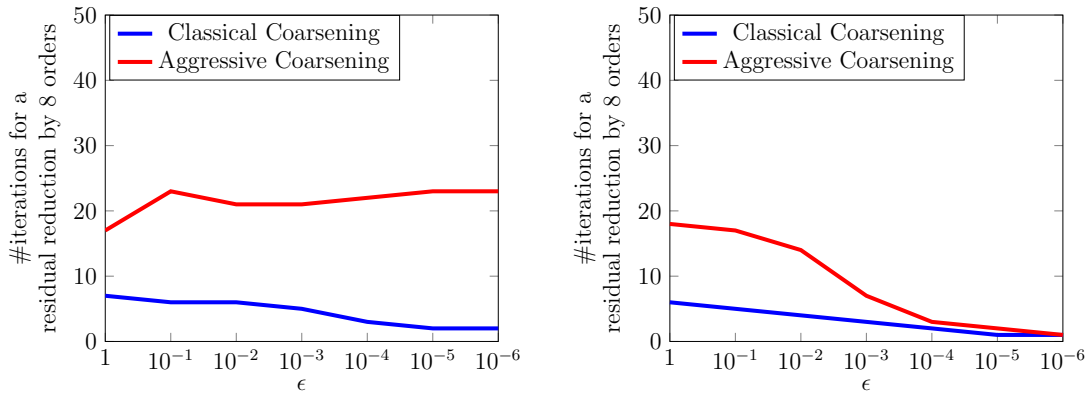


(A) $\epsilon = 1.0$ (B) $\epsilon = 10^{-3}$

FIGURE 6.3: Reduction factors of the ILU(0) smoother with different values of ϵ. With semicoarsening, high-frequency modes correspond to the blue area in the case of $H = 2h$ and to the union of the blue and green area in the case of $H = 4h$. The reduction factor that determines the smoothing factor is marked with a light blue and an orange point for $H = 2h$ and $H = 4h$, respectively.

Conclusion: With both semicoarsening and coarsening both directions the difference of the smoothing factors that result from classical and aggressive coarsening becomes smaller as the anisotropy becomes stronger. As we expect the smoothing factor to approximate the two-level convergence rate, we conclude that with strong anisotropies we can use aggressive coarsening instead of classical coarsening without negative impacts. This at least holds with ILU(0) smoothing in the model case. Figure 6.4 visualizes this with plain two-level cycles of AMG for the model problem at a grid size of $h = \frac{1}{96}$ (with $h = \frac{1}{1024}$, the second level problem is too big for a direct solver). The figure compares ILU and Gauss-Seidel smoothing

for the different coarsening variants and shows that the convergence rate between classical and aggressive coarsening with ILU-smoothing becomes the closer, the stronger the anisotropy is.



(A) Gauss-Seidel Smoothing (B) ILU(0) Smoothing

FIGURE 6.4: Number of two-level AMG iterations needed to achieve eight orders of residual reduction for the model problem with Gauss-Seidel and ILU(0) smoothing. Different values of ϵ are compared. To avoid mixing effects, AMG is used stand-alone, i.e., not as a preconditioner.

6.1.3 Relevance for the General Case

In reservoir simulations, the permeability field induces strong anisotropies. The permeability varies up to 6 orders of magnitude between neighboring grid cells in a typical simulation. For anisotropies of such strengths, our previous considerations imply aggressive coarsening to be "as good" as the classical one.

However, we have only analyzed a model problem with a constant anisotropy in one direction, whereas in reservoir simulation the permeability, and, hence, the anisotropy, varies in direction and strength virtually randomly. While the application of LFA is practically impossible for such problems, we observe comparable convergence rates for aggressive and classical coarsening with ILU(0) smoothing in practice. We can empirically further validate this observation.

Definition 6.4. Let $A = (a_{ij})$ be an arbitrary, symmetric square matrix. The graph Laplacian $GL_A = (l_{ij})$ is defined as

$$l_{ij} = \begin{cases} -1 & \text{if } a_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \neq j \text{ and } l_{ii} = \sum_{j \neq i} -l_{ij}.$$

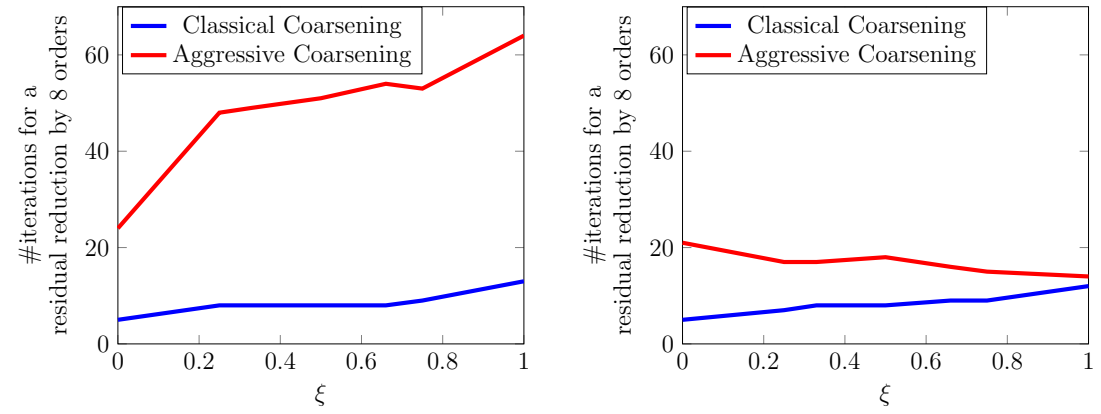
△

Let us assume a matrix A results from the discretization of an IMPES pressure equation (2.12) with Dirichlet boundary conditions at a well bore. Assume A_L to be the corresponding graph Laplacian with "analog boundaries". Clearly, A and A_L are spd M-matrices and have the same non-zero pattern. However, A_L corresponds to the discretization of a homogeneous IMPES pressure problem, while A reflects all heterogeneities from the reservoir. We can now use some $0 \leq \xi \leq 1$ and mix both matrices:

$$A_\xi := \xi A + (1 - \xi) A_L. \tag{6.8}$$

The parameter ξ allows to control the heterogeneity of the problem, while the matrix for any choice of ξ remains an spd M-matrix. Hence, we can test stand-alone Scalar-AMG with classical and aggressive coarsening for different choices of ξ and for ILU(0) or Gauss-Seidel smoothing.

From the results with representative test problems we see that the AMG con-



(A) Gauss-Seidel Smoothing (B) ILU(0) Smoothing

FIGURE 6.5: Number of iterations needed to achieve eight orders of residual reduction for A_ξ (cf. (6.8)) with Gauss-Seidel and ILU(0) smoothing. Different values of ξ are compared. Only the pressure problem is solved with the heterogeneous problem resulting from the pctst case (cf. Appendix A). To avoid mixing effects, AMG is used stand-alone, i.e., not as a preconditioner.

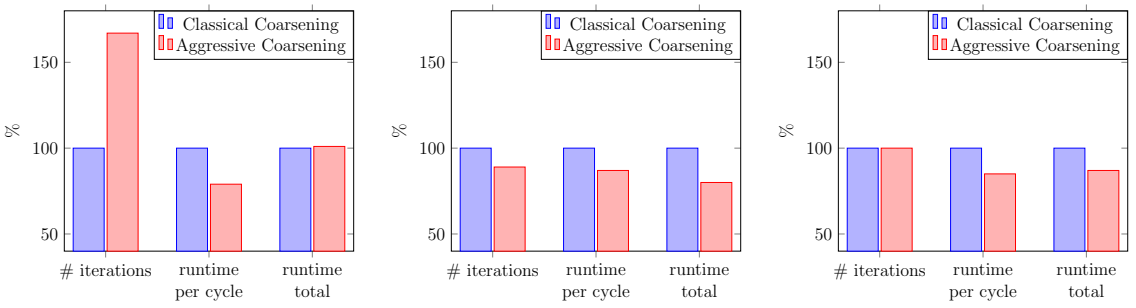
vergence for all tested values of ξ suffered when choosing aggressive coarsening, if Gauss-Seidel smoothing was used. With ILU(0) smoothing, however, the AMG convergence with aggressive and classical coarsening becomes closer with $\xi \to 1$, i.e., the more heterogeneous the problem is. This is exemplarily shown for the pctst case in Figure 6.5. This result matches what we have seen with the model problem in Figure 6.4.

6.1.4 Results

For our System-AMG approach for the solution of coupled systems from FIM simulations, for reasons of robustness, we typically use ILU(0) smoothing at the finest level. According to the previous motivations, we can then use aggressive coarsening at the finest level without reducing the convergence speed. This invalidates our initial fear that aggressive coarsening for the System-AMG approach, due to additional and expensive iterations, will not pay. On the contrary, aggressive coarsening reduces the overall run times, as demonstrated in Figure 6.6. As we use ILU smoothing only at the finest level, we only coarsen the first level aggressively. For the further levels, we proceed with standard coarsening. Apart from the aggressive coarsening, we use our standard System-AMG approach for Black-Oil problems.

The figure also shows that with homogeneous pressure problems, the number of iterations drastically increases (cf. Figure 6.6a) and we do not have performance benefits anymore. Whereas with the the original, heterogeneous pressure subblocks (Figures 6.6b and 6.6c), the number of iterations does not increase when using aggressive coarsening and the performance gain per iteration carries over to the overall run time.

In addition to the performance improvement for heterogeneous problems, the memory that is required to store the AMG hierarchy is reduced by 40-50%, compared to classical coarsening.



(A) Case pctst Graph Laplacian as A_{pp} and A_{sp}

(B) Case pctst Original A_{pp} and A_{sp}

(C) Case cputest

FIGURE 6.6: Number of iterations and runtimes in total and per cycle are compared for classical and aggressive coarsening with three different Black-Oil problems (cf. Appendix A). Relative numbers are plotted for comparability. Note that in (A) the original pressure subproblems have been replaced by the respective graph Laplacians.

6.2 Handling of Explicitly Treated Unknowns in AIM Simulations

In contrast to fully implicit (FIM) simulations, in adaptive implicit simulations (AIM, cf. Section 2.5), a decoupling of the pressure from saturation and concentration unknowns can be beneficial in terms of computational performance. For those cells i where the saturations[1] are treated explicitly in the discretization of the mass balances (2.10), so-called IMPES-cells, with $i \neq j$, the non-zero pattern of the respective matrix blocks is as follows:

$$[J]_{ii} \begin{pmatrix} * & * & \dots & * \\ * & * & & 0 \\ \vdots & & \ddots & \\ * & 0 & & * \end{pmatrix} \text{ and } [J]_{ij} = \begin{pmatrix} * & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ * & 0 & \dots & 0 \end{pmatrix}. \tag{6.9}$$

Given that no unknown from any other cell depends on the saturations of cell i, decoupling them in the diagonal block $[J]_{ii}$ would eliminate these saturations from the system. Unknowns from other cells may only depend on these saturations, if the IMPES-cell i had a non-IMPES neighbor. If, however, only IMPES-cells are neighboring the IMPES-cell[2] i, by inexpensive block-diagonal transformations, we can eliminate the saturations. This accelerates the computation and application of the ILU decomposition in the post-smoothing process at the finest level (as the saturations and concentrations are not involved in the coarse grid correction, only the ILU post-smoother becomes computationally faster). The eliminated saturations are computed once the reduced linear system is solved.

However, in order to maintain the robustness of our solver approach, we must not use decoupling methods that might have an impact on the applicability of AMG, like those from Section 4.2.3. In this section we will present two methods to exploit the diagonality of A_{ps} and A_{ss} from IMPES-cells without affecting the robustness of our System-AMG approach. That is, our System-AMG approach is ensured to be applicable and, as a matter of practical experience, we gain performance benefits.

[1] In compositional simulations, also the concentrations are treated explicitly in IMPES-cells. For reasons of readability, we just refer to saturations here and note that there might be more than three "saturation unknowns".

[2] Either the simulator supplies the information for which cells this holds, or this is analyzed on the algebraic level by checking each saturation related matrix column.

In the following, we refer to an arbitrary IMPES-cell as k, where there are dependencies on the saturations only in the diagonal block $[J]_{kk}$, whereas we refer to any other cell as l.

6.2.1 Decoupling All but One Saturations

Our approach involves two steps: We first use a right scaling that shifts the coupling to the explicitly treated saturations from the pressure to one of these saturations. We then use a left scaling to remove these couplings and eliminate all but one of these saturations from the system.

We assume the DRS transformation has already been completed and note that this cannot have introduced additional couplings to saturations from IMPES-cells, which would be different if we applied DRS after our decoupling. That is, the structure of the matrix blocks that correspond to IMPES-cells still is as in (6.9) and the pressure sub-problem is well-suited to construct an AMG hierarchy.

For our IMPES-cell k, let us assume the diagonal block $[J]_{kk}$ is of size $m \times m$ and reads as

$$[J]_{kk} := \begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & 0 \\ a_{m1} & 0 & a_{mm} \end{pmatrix}. \tag{6.10}$$

Without loss of generality[3] let us assume $a_{12} \neq 0$. We then define a right scaling by C_R (cf. (4.1)), where its k-th block $[C_R]_k$ is as in the following:

$$[\tilde{J}]_{kk} = [J]_{kk} \underbrace{\begin{pmatrix} 1 & & & & \\ & 1 & -\frac{a_{13}}{a_{12}} & \dots & -\frac{a_{1m}}{a_{12}} \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}}_{:=[C_R]_k} = \begin{pmatrix} a_{11} & a_{12} & 0 & \dots & 0 \\ a_{21} & a_{22} & -\frac{a_{22}a_{13}}{a_{12}} & \dots & -\frac{a_{22}a_{1m}}{a_{12}} \\ a_{31} & & a_{33} & & \\ \vdots & & & \ddots & \\ a_{m1} & & & & a_{mm} \end{pmatrix}. \tag{6.11}$$

Regarding cell l, we use $[C_R]_l = \mathbb{1}$.

Because this method, let us call it *saturation column eliminating (SCE)*, does not at all affect the first, i.e., pressure related column of any block, we are guaranteed

[3] If the coupling between pressure and the first saturation was zero, then the methodology works analogously with the third, fourth, etc. instead. If all these couplings already were zero, then, regarding cell k, there was no need for a decoupling.

that the pressure sub-problem is not affected. Because the pressure related column also is not involved in the scaling, no couplings to saturations from IMPES-cells are introduced in any block of the Jacobian.

The SCE method does not eliminate the explicitly treated saturations, but shifts the coupling to them from the pressure to one saturation. In order to eliminate the saturations, we need to remove these couplings. This can be achieved by a left scaling C_L, where, for our IMPES-cell k, we define:

$$[C_L]_k = \begin{pmatrix} 1 & & \\ & [\tilde{A}_{ss}]_{kk}^{-1} & \end{pmatrix}, \tag{6.12}$$

with $[\tilde{A}_{ss}]_{kk}$ the saturation sub-block of the diagonal block $[\tilde{J}]_{kk}$ from (6.11). As before, regarding cell l we set $[C_L]_l = \mathbb{1}$.
Because $[A_{ss}]_{kk}$ from the original problem is diagonal and non-singular, $[\tilde{A}_{ss}]_{kk}$ from (6.11) is an upper triangular, non-singular matrix, and $[\tilde{A}_{ss}]_{kk}^{-1}$ is easy to compute.
C_L does not modify the sub-matrix A_{pp}. As we already have applied the DRS transformation, we do not care about scalings applied to A_{sp}. These effect rows that correspond to unknowns that we will eliminate from the system.

After applying both scalings, we have decoupled all but one of cell k's saturations from the system. The decoupled unknowns do no longer occur in the linear solution process, but are updated afterwards. More precisely, for IMPES-cell k, with $j \neq k$, after the application of both scalings, we have the following non-zero structure in the respective matrix blocks:

$$[\tilde{J}]_{kk} \begin{pmatrix} * & * & 0 & \dots & 0 \\ \tilde{*} & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \tilde{*} & 0 & \dots & 0 & 1 \end{pmatrix} \text{ and } [\ \tilde{J}]_{kj} = \begin{pmatrix} * & 0 & \dots & \dots & 0 \\ \tilde{*} & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ \tilde{*} & 0 & \dots & \dots & 0 \end{pmatrix}, \tag{6.13}$$

with $\tilde{*}$ indicating some possible change compared to (6.9).
As we did not at all modify the pressure sub-matrix A_{pp}, or other sub-matrices that correspond to implicitly treated unknowns, the applicability of our System-AMG approach for the reduced system is unchanged.

6.2.2 Saturation-Decoupling in the Post-Smoother

Instead of the previous SCE decoupling prior to the application of System-AMG, we can include a decoupling, actually a simpler one, in the post-smoothing operator at the finest level. This requires less matrix scaling operations, however, it modifies the ILU fine-level smoother.

Let us assume that for our initial matrix J we have an incomplete factorization $J = LU - R$ with the respective ILU operator as

$$S_{ILU} = \mathbb{1} - U^{-1}L^{-1}J. \tag{6.14}$$

We now construct a left scaling C_{dcpl} that decouples pressure and saturations for IMPES-cell k via a Schur complement in the respective diagonal matrix blocks. This is, we compute $[C_{dcpl}]_k$ as in the qIMPES decoupling (4.8). For all cells l we simply set $[C_{dcpl}]_l = \mathbb{1}$. We define $J_{dcpl} := C_{dcpl}J$ and assume that we have the incomplete factorization $J_{dcpl} = L_{dcpl}U_{dcpl} - R_{dcpl}$.
For our initial linear system, we now modify the ILU operator as follows:

$$S_{modify} = \mathbb{1} - U_{dcpl}^{-1}L_{dcpl}^{-1}C_{dcpl}J, \tag{6.15}$$

which is equivalent to the ILU operator S_{dcpl} for the system that is described by J_{dcpl}. Hence, as long as this operation converges, our modified ILU operation for the original system does converge as well. The advantage of S_{modify}, compared to S_{ILU}, is that we can eliminate the explicitly treated saturations from the system that we need to compute the factorization for.
The p-th iteration that leads to S_{modify} then is as follows:

- For the system $Jx = f$, with the iterate x_{p-1}, compute $r_{p-1} = f - Ax_{p-1}$.
- Scale $\tilde{r}_{p-1} = C_{dcpl}^{-1}r_{p-1}$.
- Solve $L_{dcpl}U_{dcpl}e_{p-1} = \tilde{r_{p-1}}$:
  - Forwards- and backwards solution for all non-decoupled unknowns.
  - Explicit update of decoupled saturations.
- Get the new solution iterate $x_p = x_{p-1} + e_{p-1}$.

That is, we modify our ILU post-smoother in order to exploit the diagonality of A_{ps} and A_{ss} regarding explicitly treated saturations. According to practical experience, this modified smoother is working properly with our System-AMG approach.

Remark 6.5. *We have to note that the decoupled saturations must be updated after all other unknowns from the entire system are computed. It is not sufficient to wait only until the pressure of the respective cell is computed. This is because, although no other unknown depends on a decoupled saturation, any decoupled saturation may depend on further implicit unknowns that need to be updated before updating this saturation.* △

Remark 6.6. *We never need to explicitly store J_{dcpl}. The application of the scaling by C_L can be incorporated in the computation of the factorization on the fly and in-place. There is no need to consider changes in the non-zero pattern of J: all pressure related blocks in A_{pp} and A_{sp} have the same non-zero pattern, as all transmissibilities depend on each phase's pressure.* △

6.2.3 Results

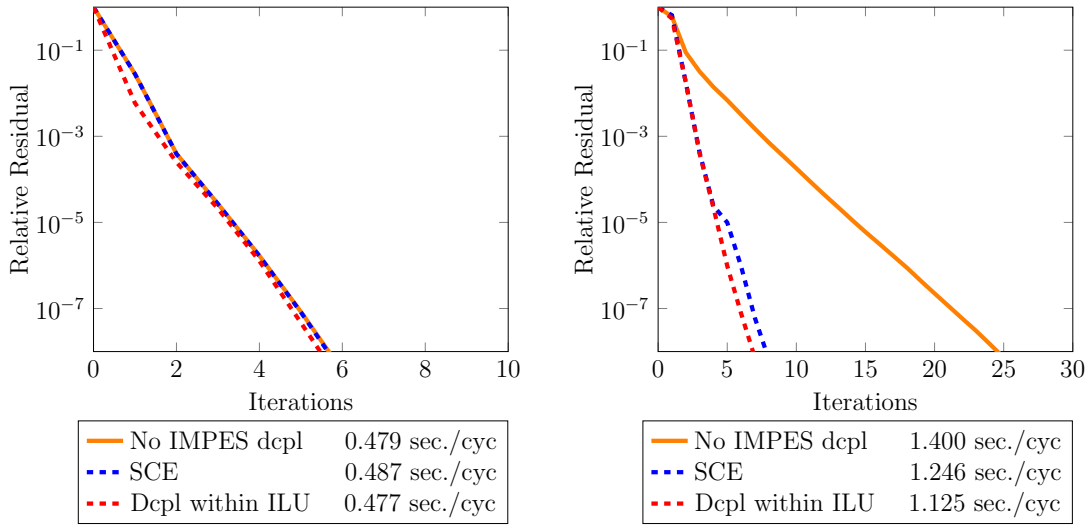
We have described two different ways to exploit the diagonality of the saturation related matrix parts from IMPES-cells in adaptive implicit simulations. Both methods have in common that the pressure related matrix part is never affected in a way that AMG might suffer from. The SCE decoupling method has the advantage of constructing an equivalent linear system that our System-AMG approach can be applied to without further modifications. However, it has the disadvantage of requiring two matrix scalings. In contrast to this, modifying only the ILU smoother involves one rather simple scaling.

According to practical experience, both methods work well with AIM problems and reduce the computational time per iteration. Figure 6.7 compares the proposed methods with not using any special handling for the explicit saturations. The systems have artificially been created from SPE10 problems (the Black-Oil and the compositional one, cf. Appendix A) by removing all off-diagonal entries in the sub-blocks of A_{ss} and A_{ps}. Apart from the methods regarding explicit saturations, our respective default System-AMG approaches are used.

While the residual reduction always is at least as good as without such a handling, the run time per AMG cycle is decreased by using the decoupling methods. This

effect is stronger with the compositional simulation, as there are more explicitly treated unknowns per IMPES-cell.

We need to discuss the drastic convergence improvement in the compositional case: In those matrix-blocks where we could decouple saturations, the reduced system is (nearly) the same for the Black-Oil and the compositional case. With both approaches for handling explicit saturations, the entries of the residual vector that correspond to decoupled saturations, as these are updated explicitly, are zero after every iteration. It does not matter whether this is a small or large number of saturations (i.e., Black-Oil or compositional). For the considered test cases this effect is over-emphasized, as using diagonal matrices for A_{ss} and A_{ps} corresponds to all cells being IMPES-cells.
In both cases we consider the SPE10 reservoir. Therefore, in the compositional case, with using the special handling of IMPES-saturations, we have nearly the same reduced system and convergence history as in the Black-Oil case. As there might be some small differences between the Black-Oil and the compositional case, for instance in the linearization, both systems still are not exactly equal.



(A) Black Oil (B) Compositional; 4 components

FIGURE 6.7: Convergence history of the System-AMG approach with different options of handling saturations and concentrations from IMPES cells. For the test the blocks A_{ss} and A_{ps} from the Dead-Oil and compositional SPE10 problem (1.1 million cells, cf. Appendix A) have been diagonalized. For comparison, the run times per cycle are given in the charts' legends.

Chapter 7

Implementational Aspects of Performance Improvements

In the previous chapter we have seen that we can apply some algorithmic modifications to our System-AMG approach in order to accelerate the computational time. Of course we can also adjust the implementation of the System-AMG components in order to obtain a better computational performance. We will discuss two such aspects in this chapter:
The first relates to the parallelization, as reservoir simulations today are hardly ever performed sequentially. Although parallelization aspects are not in the focus of this thesis, we will consider the shared-memory parallelization of ILU in some detail. We do so because the ILU post-smoother is a sensitive part of our System-AMG approach and, since it involves the full system, it is computationally expensive compared to other parts. For the other parts of our System-AMG approach, we can exploit the existing parallelization of the SAMG library.
The second aspect is the exploitation of the concrete System-AMG approach and the underlying matrix properties in the implementation of the solver. We will outline how abandoning some of System-AMG's generality can save computational time and how to realize the matrix transformations, e.g. DRS, efficiently.

7.1 Parallelization of ILU(k)

As this step involves the full linear problem, a significant portion of time is spent in the ILU post-smoothing of our System-AMG approach. Hence, the ILU smoothing also is a major bottleneck w.r.t. parallel implementations of the System-AMG approach. This is even more true as the ILU factorization is a highly sequential process. Its parallelization is not straight forward and may require compromises, a dilemma which we are going to discuss. We will provide a robust parallelization of ILU with an arbitrary, pre-defined pattern[1] $P_{\neq 0}$ on a shared memory architecture, e.g., by the OpenMP programming model [80].

We assume a number of τ independent processes, strictly speaking threads of the same process, to be running. We are not considering implicit parallelism here, which enables OpenMP to parallelize simple loops in an automatic way. Instead we focus on explicit parallelism, where the amount of automatism from OpenMP is minimized[2]. Each of the threads is independent in the sense that different operations can be executed by two threads consecutively.

We are considering some arbitrary, regular square matrix $A = (a_{ij})$ within this section. We assign each matrix row to a partition $\Omega(\mathcal{T})$, which is assigned to thread $\mathcal{T}$. In our discussion, we use the simple-most partitioning where $\Omega(1)$ consists of the matrix rows $1, ..., r_1$, $\Omega(2)$ consists of rows $r_1 + 1, ..., r_2$ and so on, until $\Omega(\tau)$ consists of rows $r_{\tau-1} + 1, ..., n$. Different partitionings could be realized by a renumbering of the matrix. We are primarily interested in the parallelization of ILU and do not discuss the, possibly beneficial, impact of such different partitionings.

There are essentially two approaches of parallelizing incomplete factorizations. An implicit renumbering of the matrix can be used that reflects the needs of concurrency. We will review this in the following and see that such a renumbering can have drastic algorithmic impacts. The other approach uses wavefronts that determine concurrency that already is inherent in the matrix. The approach we are going to use follows this direction.

[1] We can apply the same methodology with a block-ILU, and we will see that, in principle, block-ILU corresponds to an ILU(b) with a special pattern (cf. Remark 7.7).

[2] In terms of implementation, we use parallel OpenMP regions rather than only parallel do or for loops.

7.1.1 Parallelization by Renumbering

Using a multicoloring of the adjacency graph of the matrix to factorize is a popular approach to parallelize incomplete factorizations. Such a method has been introduced by Ma and Saad or Hysom and Pothem [48, 67], and still is popular today (see, for instance, Heuveline et.al. [47]). It has the advantage of being realizable for a general matrix in a rather straight forward way and typically gives a good parallel efficiency. For rather isotropic problems the convergence properties usually still are as good as in the sequential case. However, the resulting factorization generally is not equivalent to the sequential one and the ILU method may suffer from a lack of robustness. We will see that this especially holds in the highly heterogeneous problems from reservoir simulations.

7.1.1.1 Description of the Method

The idea is to assign a color to each of the matrix rows and then renumber the problem accordingly. We distinguish between inner rows and border rows, where inner rows in the ILU pattern $P_{\neq 0}$ do not have any connection to another thread's partition and border rows are all non-inner rows. We assign each inner row the color 0 and then assign the border rows some colors 1,2,..., such that two rows that are connected by some non-zero coupling do not share the same color. This can be realized by parallel graph coloring algorithms like the one proposed by Luby [66].

We can then renumber the linear system to solve, by using the rows of the first partition with color 0 first, followed by color 0 from the second partition, etc. This renumbering allows for a straight forward parallelization of the factorization and the ILU iterations.

Let us exemplarily consider the situation with two partitions and two colors. If we denote some arbitrary non-zero matrix block as $*$, the renumbered matrix reads as:

$$\begin{pmatrix} * & 0 & * & * & \dots \\ 0 & * & * & * & \dots \\ * & * & * & 0 & \dots \\ * & * & 0 & * & \dots \\ & & \vdots & & \end{pmatrix} \begin{matrix} \\ \leftarrow \text{Second partition, color 0} \\ \\ \leftarrow \text{Second partition, color 1} \\ \end{matrix} \quad . \tag{7.1}$$

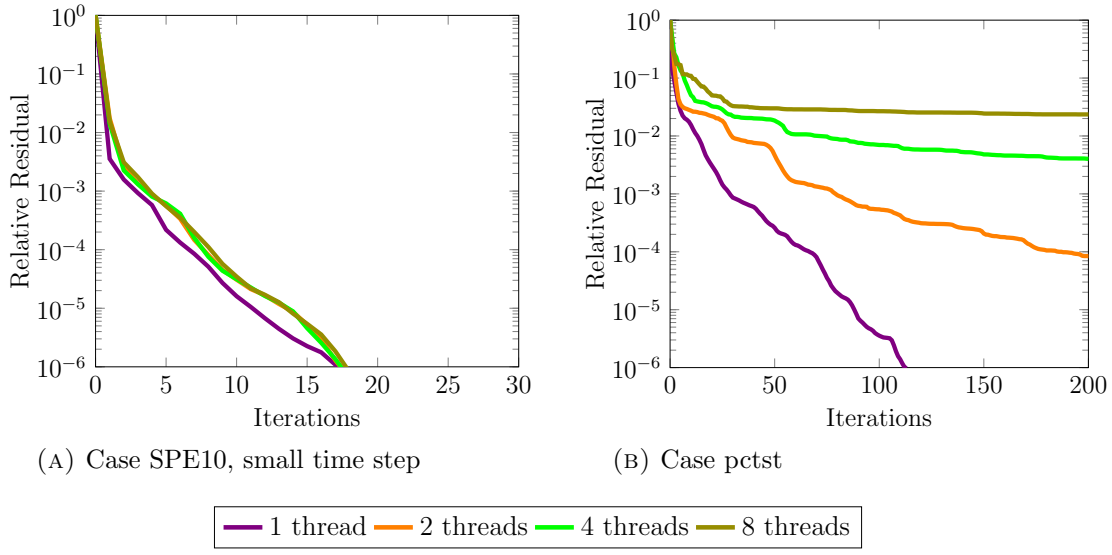
Those parts of the partitions that have the same color now can be factorized simultaneously, with a synchronization before starting the next color. The forwards and backwards solution sweeps for the systems that are described by L and U are parallelized accordingly. The proceeding, in principle, is the same with more partitions and/or colors.

7.1.1.2 Results and Drawbacks

Factorizing the renumbered matrix (7.1) is an inherently parallel task. Hence, we can expect a rather good parallel efficiency, which in fact is realized in many practical applications. However, the method suffers from a lack of convergence for many linear systems. The impacts on the convergence often highly depend on the number of partitions, i.e., the amount of concurrency, being used. The results in Figure 7.1 show these effects. For all the presented problems, ILU(0) in its sequential version can reduce the residual by six orders of magnitude in a reasonable number of iterations. However, for the cases cputest and bo8p6, already when involving two threads with the parallelization by renumbering, no convergence can be achieved anymore. For the case pctst we observe the high dependency of the solver on the number of threads.

This effect was addressed already in the early papers [48, 67]. While the linear system still describes the same problem after the renumbering, the incomplete factorization is likely not to be the same anymore. The renumbering usually results in different elements being shifted to the rest matrix of $A = LU - R$. The parallelized method then is no more equivalent to the sequential one.

A well known example for this effect are linear systems from the anisotropic Poisson problem (cf. (6.1)). With this parallelization approach, the rest matrices between different amounts of concurrency differ by $O(\epsilon)$. Hence, depending on the anisotropies' direction (i.e., with large ϵ), we have drastic impacts.
We can already see the effect with the following 4×4 matrix B, where $a, b, c \in \mathbb{R} \setminus \{0\}$ with $|a| \neq |b|$ and $|a| \neq |c|$. Let us denote R_B the rest matrix of the sequential ILU factorization of B, and R_{Bp} the rest matrix of the parallel ILU



(A) Case SPE10, small time step

(B) Case pctst

(C) Case cputest

(D) Case bo8p6

FIGURE 7.1: Convergence history of ILU(0) as single level preconditioner for flexible GMRes. The parallelization by renumbering was applied with different numbers of OpenMP threads for different full systems from Black-Oil simulations (cf. Appendix A).

factorization with two partitions. This gives:

$$B := \begin{pmatrix} a & b & c & 0 \\ b & a & 0 & c \\ c & 0 & a & b \\ 0 & c & b & a \end{pmatrix}, \; R_B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{bc}{a} & 0 \\ 0 & \frac{bc}{a} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } R_{Bp} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\frac{bc}{a} \\ 0 & 0 & 2\frac{bc}{a} & 0 \end{pmatrix}.$$

Already for this simple example, except for a different numbering, R_{B_p} is twice R_B.

7.1.2 Wavefront Parallelization

The convergence issues that we have observed with the previous parallelization approach did result from the renumbering. It was not a problem that rows have been factorized simultaneously, if their factorization only required already existing information. Hence, if the original matrix already contained groups of such rows, a parallel factorization would be possible without negative impacts on the convergence properties. While we cannot expect the "perfect" structure from (7.1), it is still likely that there are some such groups in a general matrix. This is even more true in sparse matrices resulting from the discretization of PDEs on a grid. The main task is to detect which rows are independent and which are not, especially if the underlying grid is unstructured.

7.1.2.1 Structured Grids

For problems resulting from discretizing a PDE on a structured two dimensional grid, van der Vorst [98] proposed the usage of several hyperplanes within this structured grid. We can regard them as wavefronts. If we assume the grid to be ordered lexicographically, then the method starts with factorizing the matrix rows that correspond to the $k \times k$ block of grid cells in the lower left corner of the grid. In the sense of a non-zero coupling to a row of a smaller index, these rows can only depend on rows that result from this block. Once this is finished, there are two groups of matrix rows that result from the two neighbored $k \times k$ blocks, which could be factorized independently of each other. These blocks form a level of the wavefront.

This methodology proceeds accordingly and a parallelization front, the wavefront, moves through the grid, until finally the last set of rows from the upper right block of grid cells has to be factorized by a single thread. Figure 7.2 visualizes this proceeding. The backwards and forwards sweep in ILU's solution phase are parallelized accordingly.
The principal methodology can be extended to the three dimensional case, as described by Joubert et.al. [53]. On a structured grid, this approach can provide a respectable parallel efficiency, while it is guaranteed to have the same factorization as the serial ILU method has. However, the direct transfer to a general

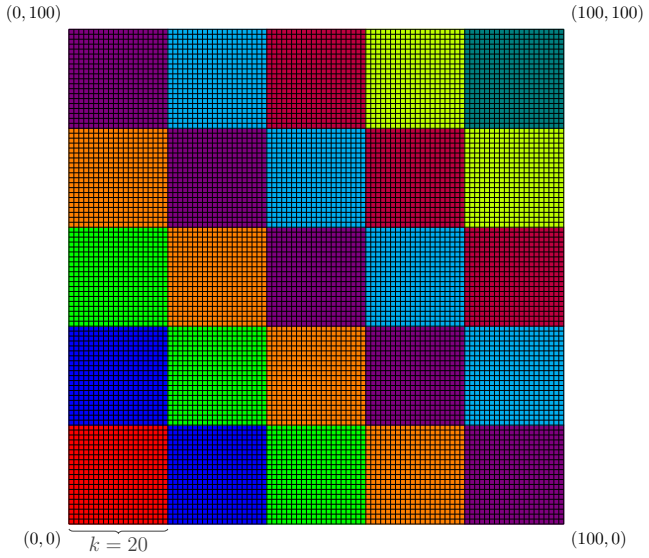


FIGURE 7.2: Wavefronts in a 2D structured grid of 100×100 cells. Each block has a size of 20×20. The blocks of the same color induce sets of matrix rows that can be factorized independently, once the sets of matrix rows with lower indices are factorized.

unstructured grid, as it is used by today's reservoir simulators, is very difficult, if not practically impossible.

7.1.2.2 Wavefronts Only in the Solution Phase

Saad [88] describes a way to exploit a wavefront, or level scheduling, approach for the solution phase of ILU that is independent of the underlying grid. The factors L and U, which are lower and upper triangular matrices, are renumbered based on a graph-depth search, such that wavefronts can directly be exploited for the backwards and forwards solution sweeps.
However, this also is the drawback of the method: the factorization itself is not parallelized at all. If ILU was used as a stand-alone solver, due to the typically high number of iterations for the considered types of linear systems, parallelizing the solution part still would be of interest (see, for instance, Naumov [76] for a recent example). However, in our System-AMG approach, we exploit AMG's numerical efficiency to keep the number of iterations small. Hence, a serial incomplete factorization would remain a serious bottleneck for the overall parallel efficiency. We therefore do not follow this approach.

7.1.2.3 Implicit Wavefronts

Inspired by the parallelization of full *LU*-factorizations for dense matrices, Dong and Cooperman [35] in 2011 presented a parallelization of the ILU(k) factorization that produces the same result as the serial version and does not exploit grid information. Their idea is to transfer the wavefronts from the grid to the matrix. They split the matrix rows into bands, which are small groups of consecutive matrix rows. Typically, these bands are smaller than our partitions. After factorizing the rows of the first band, the second band can be factorized completely. Moreover, in all further bands, the factorization can be started until a row of band two or higher is to be accessed.
With this method, there is some parallelism achieved without affecting the convergence properties in any way. However, in contrast to the initial idea of wavefronts, synchronizations are needed also while a particular row is factorized. In order to achieve a reasonable amount of concurrency, Dong and Cooperman use relatively small bands. Compared to the initial wavefront idea, this results in a drastic increase in the synchronization overhead. The severeness of this effect depends on the amount of fill-in, i.e., the pattern $P_{\neq 0}$, and on how the target hardware architecture copes with synchronizations.

7.1.2.4 Unstructured Grids - Algebraic Wavefronts

The independence of the underlying grid was one reason why algebraic multigrid is preferred to geometric multigrid in reservoir simulations (cf. Section 3.1.1), as we typically are concerned with highly unstructured grids. AMG uses a setup phase where coarse levels are constructed based on matrix information, only. For a wavefront parallelization of ILU factorizations, we will introduce a setup phase as well. It is close to an AMG setup, as we use the concept of aggregation with certain constraints on the aggregates.
Schematically, the method works as in Algorithm 7.3. We will discuss the different steps of the algorithm in detail in the following sections. The outcome of the algorithm is a set of aggregates, which are sets of row indices, where each aggregate is assigned a level[3]. The aggregates of one such level l can be factorized concurrently, as long as all rows assigned to levels of a lower index already have

[3]In our current discussion, we refer to the levels from a level-scheduling, i.e., a wavefront. This must not be mixed up with the levels of AMG's hierarchy.

Algorithm 7.3 Algebraic Wavefronts

1: **for** *each partition* T **do**
2: *Group the matrix rows of* T *into appropriate aggregates*
3: *Upwards sort the row indices of each aggregate.*
4: *Construct levels of the aggregates.*
5: *Connect levels with other partitions.*
6: *Factorize the matrix in parallel.*

been factorized. In the situation of a 2D structured grid from Section 7.1.2.1, the $k \times k$ blocks would define aggregates in the sense of the above algorithm. All of these blocks that we could factorize independently of each other, i.e., that have the same color in Figure 7.2, form a level.
We now have to discuss how to form the aggregates and levels in a general case, without using any grid information.

7.1.2.4.1 Definitions

Before we start with the description of the algebraic wavefront approach, we introduce some definitions used below.

Definition 7.1. A matrix row i *depends on a row* j if $j < i$ and $(ij) \in P_{\neq 0}$. We write $j \prec i$. △

Definition 7.2. An aggregate X *depends on another aggregate* Y, i.e. $Y \prec X$, if $\exists j \in Y, i \in X : j \prec i$. △

Definition 7.3. Let Z be an arbitrary aggregate. A *corner row* i w.r.t. Z is a row where $\exists j, k \in Z : j \prec i, i \prec k$. △

Definition 7.4. For a partition $\mathcal{T}$, a *lower partition* $\mathcal{V}$ is a partition where the indices of $\mathcal{V}$ are smaller than those of $\mathcal{T}$. That is, a row from $\mathcal{T}$ may depend on one from $\mathcal{V}$, but not vice versa. We analogously define an *upper partition* $\mathcal{W}$. △

7.1.2.4.2 Forming the Aggregates

In order to distribute the aggregates into levels later on, we need to make sure that, for any two aggregates X and Y, either $Y \prec X$ or $X \prec Y$. Hence, we do not obtain any corner aggregate.
Under this constraint, we can grow the aggregates until a given size $maxAggSize$ is reached. With this maximal size, a trade-off between the amount of concurrent

work and the need for synchronization can be adjusted. The smaller the aggregates, the easier independent aggregates can be found. But the more necessity for synchronization is expected.

The grouping of aggregates in one partition is described by Algorithm 7.4. We consider the lower and upper partitions from Definition 7.4 as one known aggregate, each. *stack* is a classical stack data structure and *low* and *high* are the lowest and highest row index of the considered partition, respectively.

Algorithm 7.4 Grouping Aggregates in One Partition

```
1: for i = low, ..., high do
2:     if i is not yet assigned an aggregate then
3:         if i is no corner row w.r.t any existing aggregate then
4:             if i only depends on rows that are assigned an aggregate then
5:                 Create aggregate X
6:                 assign i to aggregate X
7:                 k ← i
8:                 aggSize ← 1
9:                 stack ← clear()
10:                stack ← push(all j : a_ij ≠ 0; low ≤ j ≤ high;
                                   j not assigned an aggregate)
11:                while aggSize < maxAggSize and stack ≠ empty do
12:                    for all j : a_kj ≠ 0 do
13:                        if j only depends on rows assigned an aggregate then
14:                            if with j, X is no corner aggregate then
15:                                assign row j to aggregate X
16:                                aggSize ← aggSize + 1
17:                                stack ← push(all l : a_jl ≠ 0; low ≤ l ≤ high;
                                                l not yet checked and
                                                not assigned an aggregate)
18:                    k ← pop(stack)
19: for all rows i not yet assigned an aggregate do
20:     if i depends on rows from only one aggregate, Y then
21:         Assign i to aggregate Y
22:     else
23:         Create aggregate X
24:         Assign i to aggregate X
```

By using the stack structure, the algorithm attempts to construct compact aggregates, rather than "long lines". The algorithm first considers all neighbors of an aggregate's first point, before also considering the neighbors of the second point, etc. This proceeding ensures that aggregates, in the underlying adjacency graph,

grow in the direction of all couplings more or less simultaneously. We could further improve this by, for instance, monitoring the aggregate's diameter and attempting to grow aggregates such that this diameter is kept as small as possible. This additional effort, however, does not pay in practice.
As the number of rows per partition is finite, the stack is guaranteed to be empty at some point and the while-loop in the algorithm terminates. From the construction it is clear that each row index is assigned to an aggregate, in the worst case this aggregate consists of only one row index.
The smaller the aggregates, e.g., the more aggregates of size one we have, the worse we will have to expect the parallel efficiency. However, and this is the primary objective of our parallelization approach, the robustness is still exactly as in the sequential version.
We note that only data that is assigned to the considered partition is needed. Hence, a concurrent execution of the algorithm on the different involved threads is possible. The result is illustrated in Figure 7.3 for an exemplary adjacency graph.

7.1.2.4.3 Construction of Levels

We can now construct levels for the level-scheduling. This is achieved by Algorithm 7.5, where the structures $list_X$ are linked list data structures. The algorithm again is applied partition-wise, after the aggregates have been created by Algorithm 7.4.

Figure 7.3 shows the outcome of the leveling algorithm with an exemplary adjacency graph. The levels that result from Algorithm 7.5, as the method works locally per partition, do not yet capture interferences with other partitions. This way, the execution of the leveling method works concurrently. We consider the interferences between partitions later on.
To achieve a sufficient concurrency, the levels need to contain as many aggregates as possible. We do not particularly focus on this requirement in our algorithm, our primary concern is the equivalence to the sequential version. If, however, the maximal size of the aggregates is small compared to the size of the partitions, for sparse matrices it is rather likely to end with levels that contain enough aggregates to provide sufficient concurrency. We are only considering such matrices here. The algorithm has also been tested with matrices that result from bigger discretization stencils, e.g., linear elasticity, and still gave sufficient results.
Showing the well-posedness of the algorithm requires considering the two involved

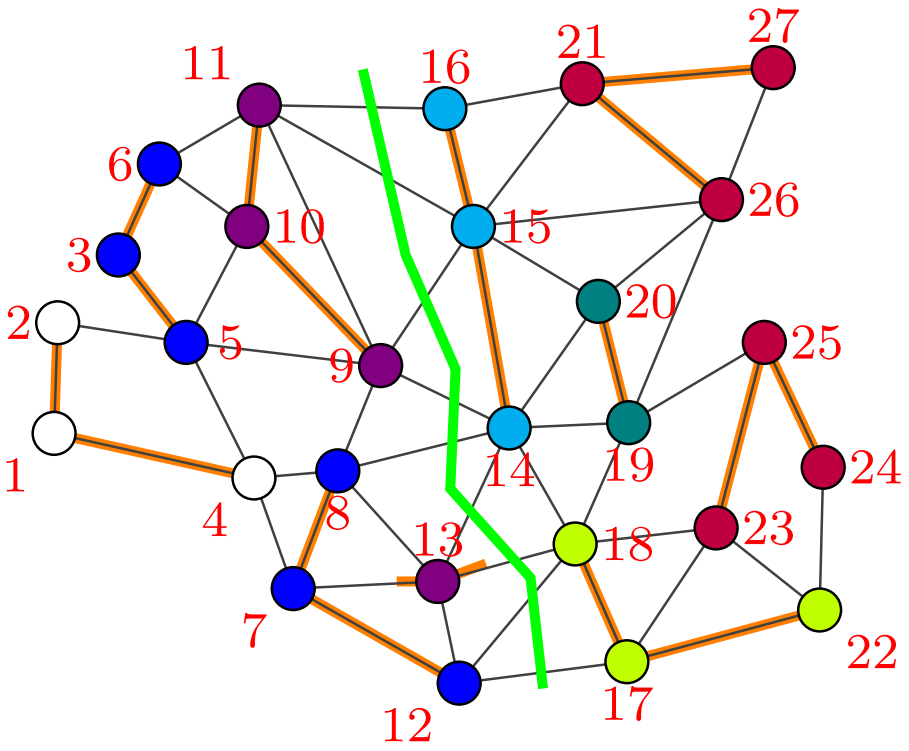


FIGURE 7.3: Visualization of the outcome of the algebraic wavefront method for an exemplary adjacency graph. The red numbers give the numbering of the respective matrix. We consider two partitions, with the green line indicating the border between them. Aggregates are illustrated by orange edges between the respective nodes. We used $maxAggSize = 3$. The color of the nodes indicates the level its aggregate belongs to. Aggregates of the same level can be factorized independently, once aggregates from lower indexed levels are factorized.

while-loops. Regarding the second one (while $list_{work} \neq empty$), we note that, due to the finiteness of the number of rows per partition, the number of aggregates per thread is finite. This implies there always is a finite number of entries in $list_{work}$. Regarding the first while-loop, we can state the following lemma, which implies the loop does terminate:

Lemma 7.5. *With aggregates created by Algorithm 7.4, $list_0$ from Algorithm 7.5, at the beginning of the first while-loop, can be empty only in the case that all aggregates have been assigned a level, i.e., the first while loop from Algorithm 7.5 terminates.*

Proof. The counter position to the above statement would mean that there is a situation where there still are aggregates without level-assignment, but all of them

Algorithm 7.5 Grouping Levels in One Partition

```
1: list_0 ← clear(); list_1 ← clear(); ....
2: for all aggregates X do
3:     get the number d(X) of aggregates it depends on
4:     list_{d(X)} ← insert(X)
5: lev ← 0
6: while ∃ aggregates not assigned a level do
7:     list_work ← list_0
8:     list_0 ← clear()
9:     lev ← lev + 1
10:    while list_work ≠ empty do
11:        X ← pop_first_element(list_work)
12:        assign aggregate X to level lev
13:        for each aggregate Y, depending on X do
14:            list_{d(Y)} ← remove(Y)
15:            d(Y) ← d(Y) − 1
16:            list_{d(Y)} ← insert(Y)
```

depend on other aggregates with not yet a level-assignment.

Let us consider the directed graph G that we construct from the aggregates without level-assignment as nodes, and their dependencies on each other giving vertices. The counter position is equivalent with this graph containing - at least one - circle. That is, with aggregates a, b and c, we have $a \prec b \prec a$, or $a \prec b \prec c \prec a$, etc. However, Algorithm 7.4 is constructed such that this is impossible. □

Remark 7.6. *In Algorithm 7.5 we need to find all aggregates Y that depend on an aggregate X. This can be implemented significantly more efficient if the pattern $P_{\neq 0}$ was structurally symmetric, i.e., $(ij) \in P_{\neq 0} \Leftrightarrow (ji) \in P_{\neq 0}$.* △

7.1.2.4.4 Connecting Levels of Different Partitions

So far, we have only handled the different partitions locally and still need to connect these information. In order to do so, for each level l that contains aggregates that depend on some lower partition, on this particular partition we find the lowest level index k where l does not depend on. These two levels can be linked, as all contained aggregates can be factorized independently of each other, once all levels with lower index on both partitions are factorized. This way, we obtain a global level structure.

7.1.2.4.5 Parallel Factorization and Solution

From the construction it is clear that the levels from the previous step can be seen

as wavefronts in the sense of Section 7.1.2.1. That is, we can step from level to level and all aggregates within one level can be factorized concurrently. We assign each of these aggregates to a thread that will compute this factorization. As the aggregates themselves are upwards ordered regarding the contained row indices, this factorization is straight forward.
The method only requires synchronizations after the completion of each level and works independently of the grid structure. We are guaranteed to obtain exactly the same factorization as we would have with the serial ILU factorization. In fact, we have only generated an instruction on how to execute exactly this factorization by multiple threads at the same time.
Finally, the backwards and forwards solution sweeps from ILU's solution phase are parallelized accordingly, completely analog to the wavefront approach for structured grids.

7.1.2.4.6 Computational Overhead of the Setup

Just as with algebraic multigrid, the setup phase comes at some overhead costs. However, with a typical sparse matrix of sufficient size, there is a huge amount of concurrency in this setup.
We also note that the approach is working with the unweighted adjacency graph of the pattern $P_{\neq 0}$. The particular matrix entries a_{ij} are irrelevant. Hence, it is likely that we can re-use an algebraic wavefront setup for several Newton iterations and time steps. Moreover, a wavefront setup for a matrix with a larger fill-in is a correct[4] result for a smaller fill-in as well. Hence, if positions for future wells are already foreseeable at the very beginning of a simulation, this can already be incorporated in the wavefront setup in order to increase its re-usability.

7.1.2.5 Results

The primary objective of our parallelization approach is maintaining ILU's robustness. We ensure that we obtain the same factorization as in serial, which inherits that there might be a coupling structure where our approach cannot provide sufficient concurrency. However, for typical reservoir simulation problems, and in fact also for other problems that show a comparable sparsity structure, we obtain acceptable parallel speedups, if we bear in mind that ILU is inherently sequential.

[4] In the sense of giving a factorizing equal to the one from the serial case.

This holds with a reasonable relation of the number of threads for the degrees of freedom.

Figure 7.4 shows performance results for representative Black-Oil problems. In

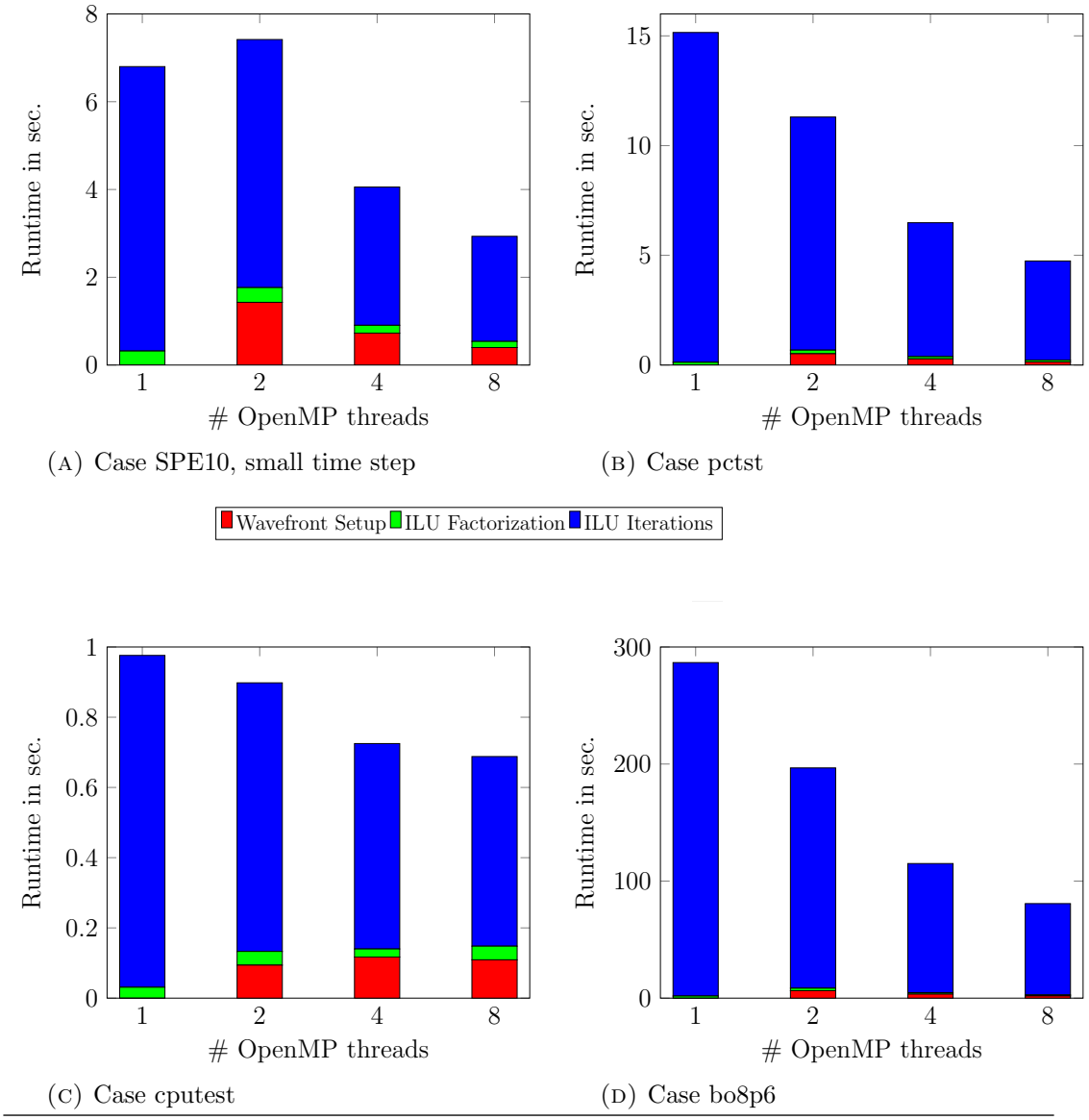


FIGURE 7.4: Performance of the Algebraic Wavefront Parallelization of ILU(0). The residual for the full systems from Black-Oil simulations (cf. Appendix A) is reduced by six orders of magnitude by using ILU(0) as a preconditioner for flexible GMRes. Since ILU(0) with any number of threads is equivalent to the serial one here, the convergence history corresponds to the single thread case from Figure 7.1.

all cases, although the construction of the wavefronts results in some overhead costs, the overall run time could be reduced by the parallelization. The smallest speedup with about 1.77 was achieved for the rather small cputest problem. Here

the construction of the wavefronts is relatively more expensive than for the bigger problems. Moreover, the small problem size limits the size of the levels that the wavefront setup is able to generate. This limits the amount of possible concurrency.
The concurrency is much better with bigger problems. For the bo8p6 problem, a speedup of 3.7 was achieved. The relatively high number of iterations was beneficial for this high speedup, as the setup costs are less significant this way. However, also for the rather well-conditioned SPE10 problem from a simulation with a small time step size, and, hence, a rather small number of 18 iterations, a speedup of 2.9 was realized, including the setup overhead of our algebraic wavefronts.

7.2 Implementational Aspects

There are two types of implementational aspects that we need to consider in order to improve the computational time. On the one hand, we do not need the full generality of System-AMG and on the other hand, we have to keep in mind that matrix transformations like DRS involve the full matrix and therefore require quite some computational time.

7.2.1 Limiting the Access to Saturation Related Data

In a general System-AMG framework all physical unknowns may be subject to some hierarchical treatment. However, in reservoir simulations we have seen that only certain physical unknowns, such as pressure, temperature and mechanical displacements, require AMG hierarchies. For the saturation and concentration unknowns it is a priori clear that they remain on the finest level. Hence, we do not need to prepare any hierarchy for them and can avoid the respective work. This holds for the following four parts of System-AMG:

1. Definition of strong couplings in the full system
2. Construction of the C/F splitting in the full system
3. Transfer operations
4. Pre-Smoothing

As there are several optimizations possible with this regard, even small benefits accumulate. Compared to using a general System-AMG realization with its full flexibility, up to 25% of computational time can be saved.

7.2.2 Matrix Transformations

In the previous chapters we have seen that matrix transformations are necessary to ensure the robust applicability of System-AMG. The transformations that we have introduced, for a point-wise ordered system matrix, are described by block-diagonal scalings. This implies that any block $[J]_{kl}$ that is zero in the original system matrix, remains zero in the transformed one. However, within non-zero blocks $[J]_{ij}$ there might be a change of the non-zero pattern (some entries of a non-zero block might be zero and which ones might change).
If we used a block data structure, this was not a problem at all. However, we then needed to deal with all non-zero blocks as dense, i.e., full, matrices. The non-zero pattern of saturation related matrix blocks is influenced from the upstream-weighting and is typically smaller than for the pressure related matrix blocks. For an exemplary SPE10 Dead-Oil system, i.e., only one saturation unknown, roughly a third of the matrix entries in a block data structure is zero. That is, we would deal with much more matrix entries than necessary, with all implications on memory requirements and performance impacts.

Instead, we could use classical CSR data structures (cf. Saad [88], Section 3.4) that are enriched with information regarding the full system (cf. [96]). Then we store only non-zero entries, even within blocks. Changes for an application of the block-diagonal matrix scalings are twofold:

- For each matrix entry in the CSR structure we need to know which block it belongs to. These information can directly be obtained from the full system information in a simple pre-processing sweep. We just need to keep in mind that, due to the disappearance of phases and components, or the presence of well equations, the sizes of the blocks might vary.
- All matrix entries that are non-zero after the application of the scaling need to be stored in the CSR structure, even if they are zero in the initial matrix. We could not properly store the scaled matrix otherwise. This, however, does not imply that we need to store all entries of non-zero blocks:

- For the DRS scaling from Section 4.4 we know that only the pressure related rows of the matrix can be changed. From the construction of DRS we also know that the non-zero pattern of the pressure related row from a block i cannot be bigger than the union of the non-zero patterns of all rows from block i.
- With our second transformation (5.7) for systems from compositional simulations under the VBF approach, we have seen that there is no additional fill-in.
- Decoupling all but one saturation unknowns of IMPES cells in AIM simulations (cf. Section 6.2.1) does, in the end, not increase the non-zero pattern of the full system matrix.

We can exploit the fact that these scalings result in a small, or even no additional fill-in, in order to minimize the number of matrix entries that System-AMG has to deal with.

Regarding the decoupling of secondary unknowns from NVF simulations (cf. Section 5.1.1.3) we note that System-AMG in fact is only used for the decoupled primary system. This requires setting up the smaller system anyway, where we can perform the decoupling block by block.

Remark 7.7. *If we should have chosen block-ILU instead of ILU as a post-smoother for our System-AMG approach, which might be necessary in some challenging simulations (cf. Figure 5.4a), we can exploit the fact that for the block-diagonal matrix scaling we already have determined the block-structure of the system.*

We can typically even use a non-block version of ILU, which we refer to as ILU(b), instead of block-ILU(0). The non-zero pattern of ILU(b) consists of all non-zero blocks. If this factorization of A, L_bU_b, exists, i.e., if the block-ILU(0) factorization L_BU_B exists and does not require pivoting for the full LU factorizations of the diagonal blocks of U_B, then, according to Lemma 3.7, ILU(b) and block-ILU(0) are equivalent. Hence, if block-ILU(0) does not require pivoting, we can see it as an ILU(k) (namely ILU(b)), just with a possibly increased non-zero pattern compared to ILU(0). If A was an M-matrix, according to Manteuffel [68], this induces block-ILU(0) to converge at least as good as ILU(0) does. △

Chapter 8

Conclusions and Outlook

In this thesis we have discussed robust AMG-based solution approaches for linear systems from fully and adaptive implicit reservoir simulations of different complexities. Already in basic Black-Oil simulations, the direct application of AMG-based approaches may fail. We have presented a preparatory matrix transformation that we used to ensure the robust applicability of AMG. This transformation shields AMG from matrix properties that might be problematic for properly constructing a hierarchy. With this transformation, no further difficulties are observed for AMG in all considered test cases.

Next to the Black-Oil case, we have discussed linear solver approaches for more sophisticated models like compositional simulations and models that take thermal and geomechanical effects into account. We have seen that the different sub-parts of the matrices, corresponding to the different types of unknowns and the couplings in-between, can have very different properties.
We have discussed how to construct robust System-AMG approaches for all of these simulation models. These approaches exploit a hierarchy wherever beneficial and this way outperform one-level solver methods. As the fluid flow within a reservoir is driven by pressure differences, we always employ a pressure hierarchy. However, also a hierarchical treatment to further types of unknowns, like temperature and mechanical displacements, is necessary in an efficient solver approach.

Only a few, typically known, information need to be made available to our System-AMG solver. It then acts in a black-box fashion and adapts itself to a concrete problem, i.e., the reservoir properties, the number of considered components, etc.

We have demonstrated the efficiency and the robustness of the System-AMG approach with challenging industrial test cases and we have been able to solve all problems that we have been concerned with.

After having described how to apply System-AMG robustly, we have discussed ways to further improve the performance. We have seen that under certain conditions, which typically are met in reservoir simulations, more aggressive coarsening variants are beneficial in terms of both memory and performance. In addition, specifically for adaptive implicit simulations, further matrix transformations can be applied that reduce the degrees of freedom and accelerate the performance.
Finally, we have introduced algebraic wavefronts as an approach to parallelize incomplete factorizations on shared memory architectures without algorithmic compromises. We have seen that, for exemplary reservoir simulation problems, this exploits parallelism and still guarantees the robustness of the sequential version.

Options for further research essentially are twofold. On the one hand, an extension of our System-AMG approaches to further simulation models is of interest. For instance, we have seen how System-AMG works when coupling fluid flow with mechanical effects from poroelasticity. However, this did not include mechanical effects like fracture propagation (i.e. fracking). We also did not cover linear systems from simulations which take chemical reactions between different components into account.
On the other hand, although it already provides an efficient and robust solver method, also the current approach might be improved further. We could, for instance, consider additional physical information from the simulator: The dynamic weights in our matrix transformation could take the concrete fluid decomposition into account. We could also employ more physical information in AMG's setup, if we used ideas of algebraic multiscale approaches [104] to further improve the interpolation of AMG. This would require the ability to exploit these ideas independently of the geometry.
Another direction of research could target at waiving the ILU smoother. Non-Galerkin AMG approaches [40] might be a way to consider the saturations in the hierarchy, but still have comparable couplings between saturations and other unknowns in the initial problem.

Appendix A

Description of Test Cases and Benchmark Environment

In this appendix we are going to describe the test cases that have been used to demonstrate certain effects within this thesis. It is meant to provide an overview on these problems in order to avoid including the description of the problems in the thesis' main part. Moreover, the technical settings for the benchmarks that are presented in this thesis will be described.

A.1 Test Cases

The test cases that we have used to investigate effects and to show results and improvements essentially have two sources. Either they are generated by the ADGPRS reservoir simulator from Stanford University's Supri-B group (see for instance [18, 110]), or they have been provided by Computer Modeling Group Ltd. (CMG) in Calgary, Canada. The geomechanical problems are taken from the Sparse Matrix Collection of the University of Florida, which is publicly available on the internet.
The proposed approaches have been used and tested also with various linear systems from other sources. However, there are either less details on the background of these problems known and / or there is currently no permission for using them within a published thesis.
The presented test cases in order are as follows:

- **Dead-Oil** problems from ADGPRS.
  There are different linear systems from the **SPE10** benchmark taken at different time steps with two phases (hydrocarbons and water). The SPE10 benchmark is the second problem from the 10^{th} SPE Comparatitive Solution Project, proposed by Christie and Blunt [24]. It consists of 1.122 million cells, which are perforated by one injection and four production wells. The permeability field is characterized by huge heterogeneities, as shown in Figure 2.2.

- **Black-Oil** problems from CMG's IMEX simulator.
  In all cases three phases (oil, water and gas) are considered in compressible models with discretizations in three dimensional, unstructured grids

  - **cputest**
    This model has 29,493 reservoir cells with relatively high aspect ratios. The reservoir is perforated by 47 wells, where the injectors inject water and gas. Nearly 1% of the reservoir cells are perforated by wells.
  - **pctst**
    Here we have 195,182 reservoir cells. The reservoir is perforated by eleven production wells in 0.06% of its cells.
  - **mx1041**
    The grid in this case is structured into ten vertical layers. It has 1,093,050 reservoir cells with a constant permeability field. Hence, heterogeneities result from the compressibility only. The reservoir is perforated by 103 wells, with injectors only injecting water. 0.02% of the reservoir cells are perforated.
  - **bo8p6**
    The reservoir is discretized with 2,638,003 cells and perforated by 1003 wells. In total 18,187 reservoir cells are perforated by a well, which is 0.7% of the cells.

- **Compositional** problems from ADGPRS under the **natural variable formulation**.

  - **Case01**
    The three dimensional model is from the ADGPRS test suite. The

discretization into 7,500 cells uses three layers in the vertical direction. The reservoir is perforated by one production well in the one corner and one gas injector at the opposite side. The model distinguishes between two phases and separates the hydrocarbon phase into nine components.

  - **SPE10 as compositional model**
    This model uses the settings from the SPE10 Dead-Oil model from above. Just that the hydrocarbon phase is separated into four different components.

- **Compositional** problems from CMG under the **volume balance formulation**.
  The problems are generated with the GEM simulator. Three phases (oil, water and gas) are used here.
  - **Case 1**
    This is a three dimensional model with eight components. The reservoir is discretized with 94,093 cells. 36 of them are perforated by two wells.
  - **Case 2**
    The three dimensional model features nine components and uses a grid of 963,536 cells. Five wells are perforating 1250 of the cells.

- **Thermal problems**
  The problems result from CMG's STARS simulator from a steam flood simulation with a reservoir that is discretized in 235,224 cells. It is perforated by 35 wells. The model considers two phases in two components.
  We use the linear systems from **two different time steps** as benchmarks.

- **Geomechanical problems**
  The systems are from the **Cube-Coup case** of the Janna group in the sparse matrix collection of the University of Florida. The model couples single phase flow with mechanical displacements in three dimensions. The underlying grid has 541,190 cells and the only difference between the two presented problems is the **time step size**, which is 10^0 and 10^6, respectively. A closer description of the problem is given by Janna et.al. [51], whereas more information on the matrix collection are provided by Davis and Hu [33].

A.2 Benchmark Environment

The results provided in the thesis have all been generated with the "oil interface" of the SAMG software [96], version 27z1 and z2. Performance results have all been measured on Intel ® Sandy Bridge computing nodes with dual octacores (E5-2660 @ 2.2GHz) and 32 GB RAM. The nodes have been running under Scientific Linux 6.4. SAMG was compiled with the Intel ® Fortran compiler ifort of version 13.0.1.
SAMG, apart from the modifications that are explicitly mentioned in this thesis, was used with its default parameters, as of version version 27z1 from January 2015. We always measure residuals in the Euclidean norm, regarding the initial (i.e. unscaled) problem.

Appendix B

Proof of Lemma 3.15

In this appendix we are going to show that $\rho(S_A^{sym}) < 1$. Here, S_A is the iteration operator of ILU(0) for solving the linear system that is described by A from Section 3.3. $S_A^{sym} := \frac{1}{2}(S_A + S_A^T)$ is the symmetric part of S_A. We note that generally S_A is not symmetric, because, although Q_A and R_A are symmetric, this does not need to hold for $S_A = Q_A^{-1}R_A$. This also implies that S_A and S_A^T do not necessarily commute and we cannot directly derive $\rho(S_A^{sym})$ from $\rho(S_A)$. However, for S_A we can exploit the following theorem.

Theorem B.1. *(Theorem 1 from [111]) Let (X,||.||,$\prec$) denote a Banach space of elements $x, y, z \in X$ with a binary relation $\prec$. Let furthermore two bounded linear operators $V, W : X \to X$ be given. If there holds:*

1. *$\prec$ is reflexive ($x \prec x$) and transitive ($x \prec y,\ y \prec z \Rightarrow x \prec z$).*
2. *$||.||$ is monotonic, i.e. $0 \prec x \prec y \Rightarrow ||x|| \leq ||y||$.*
3. *$x \prec y \Rightarrow x + z \prec y + z$.*
4. *$0 \prec x \Rightarrow 0 \prec Vx, 0 \prec Wx$.*
5. *There exists $x_0 \in X$, with $0 \prec x_0$, such that:*
   (a) *$\rho(W) = \lim_{n\to\infty} ||W^n x_0||^{\frac{1}{n}}$ and $\rho(V + W) = \lim_{n\to\infty} ||(V + W)^n x_0||^{\frac{1}{n}}$.*
   (b) *$WV^j W^k x_0 \prec A^j B^{k+1} x_0$ for $j = 1, 2, ...; k = 0, 1, ...$*

then $\rho(V + W) \leq \rho(V) + \rho(W)$. △

Theorem B.2. *For ILU(0) on a symmetric pattern, with S_A^{sym} and S_A as defined above, there is $\rho(S_A^{sym}) < 1$.*

Proof. Since each finite dimensional and normed vector space is a Banach space, $(\mathbb{R}^{n+1}, ||.||_{std1})$ is a Banach space with $||.||_{std1}$ being the classical one-norm.
We define the operator $\mathcal{T} := \begin{pmatrix} S_A & 0 \\ 0 & \rho(S_A) \end{pmatrix}$, which gives an operation from $\mathbb{R}^{n+1}$ to $\mathbb{R}^{n+1}$. We recall that ILU(0) for our A yields a regular splitting and therefore $\rho(S_A) < 1$ (cf.[71, 100]). This implies $\rho(\mathcal{T}) = \rho(\mathcal{T}^T) = \rho(S_A) < 1$.
We therefore, as all norms in a finite dimensional vector space are equivalent, conclude $\mathcal{T}$ is a bounded operator, which means $\exists M > 0 : \forall v \in \mathbb{R}^{n+1} : ||\mathcal{T}v||_{std1} \leq M||v||_{std1}$. That is, $\mathcal{T}$ meets the requirements from the previous theorem.
We define the relation $\prec$ as: $x \prec y \Leftrightarrow \forall i \in \{1, .., n+1\} : x_i \leq y_i$ and continue with the further conditions:

1. The reflexivity and transitivity of $\prec$ are obvious.
2. If $0 \prec x \prec y$, we have $0 \leq x_i \leq y_i$ for all i.
   This directly implies $||x||_{std1} \leq ||y||_{std1}$.
3. Follows directly from the element-wise construction of $\prec$.
4. Since A is an M-matrix, from Meijerink [71] we know ILU(0) to result in a regular splitting. Hence, Q_A^{-1} and R_A are non-negative matrices. Consequently, $S_A = Q_A^{-1}R_A$ is non-negative as well, and so is $\mathcal{T}$. Since $0 \prec x$ yields all entries of x to be non-negative, there is no way for any entry of $\mathcal{T}x$ to be negative. Hence, $0 \prec \mathcal{T}x$. The same holds with $\mathcal{T}^T$.
5. For any $\xi \in \mathbb{R}$ there is $(\mathcal{T})^\xi = \begin{pmatrix} (S_A)^\xi & 0 \\ 0 & \rho(S_A)^\xi \end{pmatrix}$. The same analogously holds again for $\mathcal{T}^T$. Now, let $x_0 = e_{n+1} = (0...01)^T$, then there is:
   (a) $\lim\limits_{n\to\infty} ||(\mathcal{T})^n x_0||^{\frac{1}{n}} = \lim\limits_{n\to\infty} || \begin{pmatrix} (S_A)^n & 0 \\ 0 & \rho(S_A)^n \end{pmatrix} x_0||^{\frac{1}{n}} = \lim\limits_{n\to\infty} ||\rho(S_A)^n||^{\frac{1}{n}} = \rho(S_A) = \rho(\mathcal{T})$. The same analogously holds for $\mathcal{T}^T$ and $\mathcal{T} + \mathcal{T}^T$.
   (b) For any $j = 1, 2, ...; k = 0, 1, ...$ there is:
   $\mathcal{T}(\mathcal{T}^T)^j(\mathcal{T})^k x_0 = \rho(S_A)^{j+k+1} = (\mathcal{T}^T)^j(\mathcal{T})^{k+1}x_0$

In the end, all conditions of Theorem B.1 hold and we have $\rho(\mathcal{T} + \mathcal{T}^T) \leq \rho(\mathcal{T}) + \rho(\mathcal{T}^T) < 2$. This directly implies $\rho(S_A + S_A^T) < 2$ and hence $\rho(S_A^{sym}) < 1$. □

Appendix C

Relevant Reduction Factors for the ILU(0) Smoothing Factor

In this appendix, we give the proof of Lemmas 6.2 and 6.3. That is, we show that the error reduction factors at three combinations (μ, ν), respectively, determine the smoothing factor. In Lemma 6.1, we have already found that we only need to consider the boundary of $\Phi \setminus \Phi_{low}$. Hence, in this appendix, we investigate one part of this boundary after the other. We do so for uniform- and semicoarsening. In the following discussion, we use the following well known trigonometrical relations that hold for all $\alpha, \beta \in \mathbb{R}$:

$$
\begin{aligned}
&sin(\alpha) = -sin(\alpha \pm \pi) \text{ and } cos(\alpha) = -cos(\alpha \pm \pi) \\
&sin(\alpha) = -sin(-\alpha) \text{ and } cos(\alpha) = cos(-\alpha) \\
&sin^2(\alpha) + cos^2(\alpha) = 1 \\
&sin(\alpha \pm \beta) = sin(\alpha)cos(\beta) \pm sin(\beta)cos(\alpha) \\
&cos(\alpha \pm \beta) = cos(\alpha)cos(\beta) \mp sin(\alpha)sin(\beta).
\end{aligned}
$$

We assume all definitions that we have made in Section 6.1.2.2 to hold, in particular:

$$
\begin{aligned}
\lambda_{ILU}(\mu, \nu, \epsilon, h) &= \frac{|cos((\mu - \nu)h\pi)|}{|1 + \epsilon - \epsilon cos(\mu h\pi) - cos(\nu h\pi) + \delta cos((\mu - \nu)h\pi)|} \\
d(\epsilon, h, \mu, \phi) &= \frac{1 + \epsilon - \epsilon cos(\mu h\pi) - cos((\mu - \phi)h\pi)}{cos(\phi h\pi)}.
\end{aligned}
\tag{C.1}
$$

We moreover have $c_1 := c(h, H, 1, \epsilon)$ and $c_2 := c(h, H, \epsilon, 1)$, with

$$c(h, H, \omega, \tau) := \frac{1}{H} - \frac{1}{h\pi} arcsin(\omega \frac{sin(\frac{h}{H}\pi)}{1 + \epsilon - \tau cos(\frac{h}{H}\pi)}). \tag{C.2}$$

This definition is a result of the following discussion.

C.1 Uniform Coarsening

This corresponds to Lemma 6.2 and we have $\Phi_{high} = \Phi \setminus \Phi_{low}$ with $\Phi_{low} = [1, \frac{1}{H})^2$.

First Part of the Boundary

We keep $\nu = 1$ fixed and define:

$$d_A(\mu) := d(\mu, \mu - 1) = \frac{1 + \epsilon - \epsilon cos(\mu h\pi) - cos(h\pi)}{cos((\mu - 1)h\pi)}.$$

The derivative reads as:

$$\frac{\partial}{\partial \mu} d_A(\mu) = h\pi \frac{\epsilon sin(\mu h\pi) cos((\mu - 1)h\pi) + sin((\mu - 1)h\pi)(1 + \epsilon - \epsilon cos(\mu h\pi) - cos(h\pi))}{cos^2((\mu - 1)h\pi)}.$$

Because $0 < \mu < \frac{1}{h}$, we have:

$$\begin{aligned} sin(\mu h\pi) &\in (0, 1], \\ cos((\mu - 1)h\pi) &\in \begin{cases} [0, 1] & \text{for } \mu \leq \frac{1}{2h} + 1 \\ (-1, 0) & \text{else} \end{cases}, \\ sin((\mu - 1)h\pi) &\in [0, 1] \text{ and} \\ (1 + \epsilon - \epsilon cos(\mu h\pi) - cos(h\pi)) &> 0. \end{aligned}$$

We conclude that

- For $0 < \mu \leq \frac{1}{2h}$: $\frac{\partial}{\partial \mu} d_A(\mu) > 0$
- $\lim\limits_{\mu \to \frac{1}{2h} + 1} \frac{\partial}{\partial \mu} d_A(\mu) \to \infty$

- For $\frac{1}{2h} + 1 < \mu < \frac{1}{h}$ we have

$$cos((\mu - 1)h\pi) > cos(\mu h\pi)$$
$$sin((\mu - 1)h\pi) > sin(\mu h\pi).$$

  Therefore, we we have

$$\begin{aligned}&\epsilon sin(\mu h\pi)cos((\mu-1)h\pi) + sin((\mu-1)h\pi)(1+\epsilon-\epsilon cos(\mu h\pi) - cos(h\pi))\\ &> \epsilon sin(\mu h\pi)cos(\mu h\pi) + sin(\mu h\pi)(1-cos(h\pi)) + \epsilon sin(\mu h\pi) - \epsilon sin(\mu h\pi)cos(\mu h\pi)\\ &> \epsilon sin(\mu h\pi) > 0,\end{aligned}$$

  which yields $\frac{\partial}{\partial \mu} d_A(\mu) > 0$.

Hence, extremal values for $d_A(\mu)$ with $\frac{1}{H} \leq \mu < \frac{1}{h}$ must be at $\mu = \frac{1}{h} - 1$ or at $\mu = \frac{1}{H}$. For the pair (μ, ν) this means either at $(\frac{1}{h} - 1, 1)$ or at $(\frac{1}{H}, 1)$.

Second Part of the Boundary

We now keep $\mu = \frac{1}{h} - 1$ fixed and define $\xi := 1 + \epsilon - \epsilon cos((\frac{1}{h} - 1)h\pi) = 1 + \epsilon + \epsilon cos(-h\pi)$. We can then define

$$d_B(\nu) := d(\frac{1}{h} - 1, \frac{1}{h} - 1 - \nu) = \frac{\xi - cos(\nu h\pi)}{cos((\frac{1}{h} - 1 - \nu)h\pi)} = \frac{\xi - cos(\nu h\pi)}{-cos(-(1+\nu)h\pi)}.$$

Computing the derivative w.r.t. ν yields:

$$\begin{aligned}
\frac{\partial}{\partial\nu}d_B(\nu) &= \frac{-h\pi sin(\nu h\pi)cos(-\nu h\pi - h\pi) + h\pi sin(-\nu h\pi - h\pi)(\xi - cos(\nu h\pi))}{cos^2(-\nu h\pi - h\pi)} \\
&= \frac{h\pi sin(-\nu h\pi)cos(\nu h\pi + h\pi) + h\pi sin(-\nu h\pi - h\pi)(\xi - cos(\nu h\pi))}{cos^2(\nu h\pi + h\pi)} \\
&= \frac{h\pi sin(-\nu h\pi)(cos(\nu h\pi)cos(h\pi) - sin(\nu h\pi)sin(h\pi))}{cos^2(\nu h\pi + h\pi)} \\
&\quad + \frac{h\pi\xi sin(-\nu h\pi - h\pi) - h\pi cos(\nu h\pi)(sin(-\nu h\pi)cos(h\pi) - sin(h\pi)cos(-\nu h\pi))}{cos^2(\nu h\pi + h\pi)} \\
&= \frac{h\pi sin(-\nu h\pi)sin(-\nu h\pi)sin(h\pi) + h\pi\xi sin(-\nu h\pi - h\pi)}{cos^2(\nu h\pi + h\pi)} \\
&\quad - \frac{h\pi cos(-\nu h\pi)(-sin(h\pi)cos(-\nu h\pi))}{cos^2(\nu h\pi + h\pi)} \\
&= \frac{h\pi sin(h\pi)(sin^2(-\nu h\pi) + cos^2(-\nu h\pi)) + h\pi\xi sin(-\nu h\pi - h\pi)}{cos^2(\nu h\pi + h\pi)} \\
&= \frac{h\pi sin(h\pi) - h\pi\xi sin(\nu h\pi + h\pi)}{cos^2(\nu h\pi + h\pi)}.
\end{aligned}$$

We have $0 < h < 1$, which implies $\xi > 1$. Therefore, and because $1 \leq \nu < \frac{1}{h}$, we have $\xi sin(\nu h\pi + h\pi) > sin(h\pi)$. This yields $\frac{\partial}{\partial\nu}d_B(\nu) < 0$ and we can find extremal values only for $\nu = 1$ and $\nu = \frac{1}{h} - 1$. As we have $\mu = \frac{1}{h} - 1$, the first option is identical with what we already found above.
Regarding the second option we note that

$$\begin{aligned}
\lambda_1 := \lambda_{ILU}(\frac{1}{h} - 1, \frac{1}{h} - 1, \epsilon, h) &= \frac{\delta}{|(1+\epsilon)(1 - cos((\frac{1}{h} - 1)h\pi)) + \delta|} \\
&= \frac{\delta}{|(1+\epsilon)(1 + cos(h\pi)) + \delta|} \text{ and} \\
\lambda_2 := \lambda_{ILU}(\frac{1}{h} - 1, 1, \epsilon, h) &= \frac{\delta|cos((\frac{1}{h} - 2)h\pi)|}{|1 + \epsilon - \epsilon cos((\frac{1}{h} - 1)h\pi)) - cos(h\pi) + \delta cos((\frac{1}{h} - 2)h\pi)|} \\
&= \frac{\delta|cos(2h\pi)|}{|1 + \epsilon + \epsilon cos(h\pi)) - cos(h\pi) - \delta cos(2h\pi)|}.
\end{aligned}$$

Now, in the limit case of $h = 0$ we directly have $\lambda_1 < \lambda_2$. We can therefore conclude that for sufficiently small h, and here we are interested in small h, we do not need to consider the μ, ν combination $(\frac{1}{h} - 1, \frac{1}{h} - 1)$.

Third and Fourth Part of the Boundary

So far, we have discussed two parts of the boundary of Φ_{high} that are part of the boundary of Φ ($\nu = 1$ and $\mu = \frac{1}{h} - 1$). The further two such parts of the boundary, i.e., keeping first $\mu = 1$ and then $\nu = \frac{1}{h} - 1$ fixed, follow completely analogously. As $cos((\mu - \nu)h\pi) = cos((\nu - \mu)h\pi)$, the situation is simply mirrored and we find the respective combinations of μ and ν as the transpose of what we already had. However, we can withdraw these transpose combinations from our discussion. For the case $\epsilon = 1$ we directly have $\lambda_{ILU}(\mu, \nu, 1, h) = \lambda_{ILU}(\nu, \mu, 1, h)$ and the transpose combinations would not give any new information.
In the case $\epsilon < 1$, let us consider $(\mu, \nu) = (1, \frac{1}{h} - 1)$ as the transpose of $(\frac{1}{h} - 1, 1)$. Analogously to λ_2 above, we find

$$\lambda_3 := \lambda_{ILU}(1, \frac{1}{h} - 1, \epsilon, h) = \frac{\delta |cos(2h\pi)|}{|1 + \epsilon - \epsilon cos(h\pi)) + cos(h\pi) - \delta cos(2h\pi)|}.$$

Because $\epsilon < 1$, we directly find $\lambda_3 < \lambda_2$ and we therefore do not need to consider λ_3. The same argumentation applies to $(\frac{1}{H}, 1)$ as the transpose of $(\frac{1}{H}, 1)$.

Fifth and Sixth Part of the Boundary

We still need to consider those two parts of the boundary of Φ_{high} that are part of $\partial\Phi^2_{low}$, but not of $\partial\Phi^2$. That is, we first of all keep $\mu = \frac{1}{H}$ fixed and vary ν between 1 and $\frac{1}{H}$. The situation for keeping ν fixed and varying μ accordingly again is mirrored and is therefore not explicitly discussed here.
With our fixed choice of μ we define

$$d_C(\nu) := d(\frac{1}{H}, \frac{1}{H} - \nu) = \frac{1 + \epsilon - \epsilon cos(\frac{h}{H}\pi) - cos(\nu h\pi)}{cos(\frac{h}{H}\pi - \nu h\pi)}.$$

To find the extremal values, we compute the derivative:

$$\begin{aligned}\frac{\partial}{\partial\nu}d_C(\nu) &= \frac{h\pi sin(\nu h\pi)cos(\frac{h}{H}\pi-\nu h\pi)-h\pi sin(\frac{h}{H}\pi-\nu h\pi)(1+\epsilon-\epsilon cos(\frac{h}{H}\pi)-cos(\nu h\pi))}{cos^2(\frac{h}{H}\pi-\nu h\pi)}\\ &= h\pi(\frac{sin(\nu h\pi)cos(\frac{h}{H}\pi)cos(\nu h\pi)+sin^2(\nu h\pi)sin(\frac{h}{H}\pi)+sin(\frac{h}{H}\pi)cos^2(\nu h\pi)}{cos^2(\frac{h}{H}\pi-\nu h\pi)}\\ &+\frac{-sin(\nu h\pi)cos(\frac{h}{H}\pi)cos(\nu h\pi)-sin(\frac{h}{H}\pi-\nu h\pi)(1+\epsilon-\epsilon cos(\frac{h}{H}\pi))}{cos^2(\frac{h}{H}\pi-\nu h\pi)})\\ &= h\pi\frac{sin(\frac{h}{H}\pi)-sin(\frac{h}{H}\pi-\nu h\pi)(1+\epsilon-\epsilon cos(\frac{h}{H}\pi))}{cos^2(\frac{h}{H}\pi-\nu h\pi)}.\end{aligned}$$

Due to $\epsilon > 0$, we have $1+\epsilon-\epsilon cos(\frac{h}{H}\pi) \neq 0$. We find $\nu = \frac{1}{H}-\frac{1}{h\pi}arcsin(\frac{sin(\frac{h}{H}\pi)}{1+\epsilon-\epsilon cos(\frac{h}{H}\pi)}) = c_1$ and need to consider the μ,ν combination $(\frac{1}{H}, c_1)$. A simple computation yields that we can exclude $(\frac{1}{H},\frac{1}{H})$.

Summary

The smoothing factor, which is the maximal reduction factor for high-frequency error modes, is the maximum of $\lambda_{ILU}(\mu,\nu,\epsilon,h)$ taken over the following three values of (μ,ν):

- $(\mu,\nu)_1 := (\frac{1}{h}-1, 1)$
- $(\mu,\nu)_2 := (\frac{1}{H}, c_1)$
- $(\mu,\nu)_3 := (\frac{1}{H}, 1)$

C.2 Semicoarsening

This corresponds to Lemma 6.3 and we have $\Phi_{high} = \Phi \setminus \Phi_{low}$ with $\Phi_{low} = [1,\frac{1}{h}) \times [1,\frac{1}{H})$.
We can make use of what we have found above, which shortcuts our discussion regarding the four parts of $\partial\Phi_{high}$.

We start with keeping $\nu = \frac{1}{H}$ fixed. Completely analogously as in C.1, we find that the only extremal value of the resulting function of μ within $[1,\frac{1}{h})$ can be found at $\mu = c_2$. The two boundary values $\mu = 1$ and $\mu = \frac{1}{h}-1$ also need to

be considered. These already are the three combinations from Lemma 6.3 and we now need to show that these are the only combinations to consider.

We continue with the second part of the boundary and keep $\mu = \frac{1}{h} - 1$ fixed. We then need to consider d_B from C.1. We know that there is no extremal value for $\nu \in [1, \frac{1}{h})$, which especially includes $\nu \in [\frac{1}{H}, \frac{1}{h})$.
Our argumentation regarding $\lambda_{ILU}(\frac{1}{h} - 1, \frac{1}{h} - 1, \epsilon, h) < \lambda_{ILU}(\frac{1}{h} - 1, 1, \epsilon, h)$ analogously yields $\lambda_{ILU}(\frac{1}{h} - 1, \frac{1}{h} - 1, \epsilon, h) < \lambda_{ILU}(\frac{1}{h} - 1, \frac{1}{H}, \epsilon, h)$. Hence, no additional combination (μ, ν) needs to be considered.

For the third part of the boundary, we keep $\mu = 1$ fixed. For the resulting function of ν, analogously as in C.1, we find that an extremal value is located only at $\nu = 1 - \frac{1}{h\pi} arcsin(\frac{sin(h\pi)}{1+\epsilon-\epsilon cos(h\pi)})$. However, we have $h < 1$ and, hence, $sin(h\pi) > 0$, which directly implies $\nu < 1$. As we are only considering $\nu \geq \frac{1}{H} > 1$, there are only the values $\nu = \frac{1}{H}$ and $\nu = \frac{1}{h} - 1$ to consider from this part of the boundary. The first value already is covered from the first part of the boundary. Analogously to $\lambda_{ILU}(1, \frac{1}{h} - 1, \epsilon, h) < \lambda_{ILU}(\frac{1}{h} - 1, 1, \epsilon, h)$ from C.1, we have $\lambda_{ILU}(1, \frac{1}{h} - 1, \epsilon, h) < \lambda_{ILU}(\frac{1}{h} - 1, \frac{1}{H}, \epsilon, h)$.

Regarding the remaining part of the boundary, i.e., keeping $\nu = \frac{1}{h} - 1$ fixed, according to C.1, the respective reduction factors λ_{ILU} are smaller than for the above second part of the boundary, i.e., keeping $\mu = \frac{1}{h} - 1$ fixed.

In summary, the smoothing factor of ILU(0) is given by the maximal reduction factor $\lambda(\mu, \nu, \epsilon, h)$ taken over the three values of (μ, ν):

- $(\mu, \nu)_4 := (\frac{1}{h} - 1, \frac{1}{H})$
- $(\mu, \nu)_5 := (c_2, \frac{1}{H})$
- $(\mu, \nu)_6 := (1, \frac{1}{H})$.

Bibliography

[1] Aarnes, J,E.; Gimse, T.; Lie, K.-A.: An Introduction to the Numerics of Flow in Porous Media using Matlab *Geometric Modeling, Numerical Simulation and Optimization*, pp 265-306, 2007

[2] Aavatsmark, I.: Multipoint Flux Approximation Methods for Quadrilateral Grids *9th International Forum on Reservoir Simulation, Abu Dhabi*, December 2007

[3] Aavatsmark, I.: An Introduction to Multipoint Flux Approximations for Quadrilateral Grids *Computational Geosciences* 6, pp 405-432, 2002

[4] Acs, G.; Doleschall, S.; Farkas, E.: General Purpose Compositional Model *SPE Journal*, pp 543-553, August 1985

[5] Al-Shaalan, T.M.; Dogru, A.H.; Klie, H.; Wheeler, M.F.: Studies of Robust Two-Stage Preconditioners for the Solution of Fully Implicit Multiphase Flow Problems *Saudi Aramco Journal of Technology*, Spring 2009, pp 2-11

[6] Aziz, K; Durlofsky, L.J.; Tchelepi, H.A.: Notes on Reservoir Simulation *Stanford University*, July 2005

[7] Aziz, K.; Settari, A.: Petroleum Reservoir Simulation *Applied Science Publishers*, 1979

[8] Bai, M.; Meng, F.; Elsworth, D.; Roegiers, J.-C.: Analysis of Stress-dependent Permability in Nonorthogonal Flow and Deformation Fields *Rock Mechanics and Rock Engineering*, 32 (3), pp 195-219, 1999

[9] Bank, R.E.; Chan, T.F.; Coughran, W.M.; Smith, R.K.: The Alternate Block Factorization Procedure for Systems of Partial Differential Equations *BIT Numerical Mathematics*, 29 (4), pp 938-954, 1989

[10] Batycky, R.P.; Thiele, M.R.; Blunt, M.J.: A Streamline-Based Reservoir Simulation of the House Mountain Waterflood *Stanford Center for Reservoir Forecasting* Annual Report, 1997

[11] Béhar, E.; Mougin, P.; Pina, A.: Integration of Asphaltenes Flocculation Modeling into Athos Reservoir Simulator *Oil & Gas Science and Technology*, 58 (6), pp 637-646, 2003

[12] Bergamaschi, L.; Ferronato, M.; Gambolati, G.: Mixed Constraint Preconditioners for the Iterative Solution of FE Coupled Consolidation Equations *Journal of Computational Physics* 227, pp 9885-9897, 2008

[13] Bo, L.; Wheeler, M.F.: Iterative Coupling Reservoir Simulation on High Performance Compters *Petroleum Sciences* 6, pp 43-50, 2009

[14] Brandt, A., Livne, O.E.: Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, Revised Edition *SIAM Classics in Applied Mathematics*, 2011

[15] Brandt, A.: Algebraic Multigrid Theory: the Symmetric Case *Applied Mathematics and Computation*, 19 (1-4), pp 23-56, July 1986

[16] Brandt, A.: Multi-Level Adaptive Solutions to Boundary-Value Problems *Mathematics of Computation*, 31, pp 333-390, 1977

[17] Brown, G.L.; Collins, D.A.; Chen, Z.: Efficient Preconditioning for Algebraic Multigrid and Red-Black Ordering in Adaptive-Implicit Black-Oil Simulations *SPE Reservoir Simulation Symposium*, February 2015

[18] Cao, H.: Development of Techniques for General Purpose Simulators *PhD thesis, Stanford University* June 2002

[19] Cao, H.; Tchelepi, H.A.; Wallis, J.R.; Yardumian, H.E.: Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation *SPE Annual Technical Conference and Exhibition*, October 2005

[20] Cardoso, M.A.; Durlofsky, L.J.: Linearized Reduced-Order Models for Subsurface Flow Simulation *Journal of Computational Physics*, 229 (2010), pp 681-700, 2009

[21] Chen, Z.; Huan, G.; Ma, Y.: Computational Methods for Multiphase Flows in Porous Media *SIAM*, April 2006

[22] Chen, Q.-Y.; Wan, J.; Yang, Y.; Mifflin, R.T.: Enriched Multi-Point Flux Approximation for General Grids *Journal of Computational Physics*, 227 (3), pp 1701-1721, January 2008

[23] Chen, Z.; Zhang, Y.: Well Flow Models for Various Numerical Methods *Int. Journal of Numerical Analysis and Modeling*, 6 (3), pp 375-388, 2009

[24] Christie, M.A.; Blunt, M.J.: Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques *SPE Reservoir Simulation Symposium*, February 2001

[25] Cleary, A.J.; Falgout, R.D.; Henson, V.E.; Jones, J.E.; Manteuffel, T.A.; McCormick, S.F.; Miranda, G.N.; Ruge, J.W.: Robustness and Scalability of Algebraic Multigrid *SIAM Journal on Scientific Computing*, 21 (5), pp 1886-1908, 1998

[26] Clees, T.: AMG Strategies for PDE Systems with Applications in Industrial Semiconductor Simulation *PhD thesis, University of Cologne, Shaker Verlag*, 2005

[27] Clees, T.; Ganzer, L.: An Efficient Algebraic Multigrid Solver Strategy for Adaptive Implicit Methods in Oil Reservoir Simulation *SPE Journal*, 15, 2010

[28] Coats, B.K.; Fleming, G.C.; Watts, J.W.; Ramé, M.; Shirakar, G.S.: A Generalized Wellbore and Surface Facility Model, Fully Coupled to a Reservoir Simulator *SPE Reservoir Evaluation & Engineering*, pp 132-142, April 2004

[29] Coats, K.H.: A Note on IMPES and some IMPES-Based Simulation Models *SPE Journal*, 5 (3), pp 245-251, September 2000

[30] Coats, K.H.: IMPES Stability: Selection of Stable Timesteps *SPE Journal*, pp 181-187, June 2003

[31] Collins, D.; Nghiem, L.; Li, Y.; Grabenstetter, J.: An Efficient Approach to Adaptive-Implicit Compositional Simulation with an Equation-of-State *SPE Reservoir Engineering*, pp 259-264, May 1992

[32] Coussy, O.: Poromechanics *John Wiley & Sons*, March 2004

[33] Davis, T.A.; Hu, Y.: The University of Florida Sparse Matrix Collection *ACM Transactions on Mathematical Software*, 38 (1), pp 1-25, 2011

[34] Detournay, E.; Cheng, A. H.-D.: Fundamentals of Poroelasticity *Appeared in: ed. Fairhust, C.: Comprehensive Rock Engineering: Principles, Practices and Projects, Volume II, Analysis and Design Method, Pergamon Press, 1993*, Chapter 5, pp 113-171

[35] Dong, X.; Cooperman, G.: A Bit-Compatible Parallelization for ILU(k) Preconditioning *Proceedings of the* 17^{th} *EuroPar, Springer Verlag*, pp 66-77, 2011

[36] Dung. T.Q.: Coupled Fluid Flow-Geomechanics Simulations Applied to Compaction and Subsidence Estimation in Stress Sensitive & Heterogeneous Reservoirs *PhD Thesis, University of Adelaide*, 2007

[37] Econimides, M.J.; Hill, A.D.; Ehlig-Econimides, C.; Zhu, D.: Petroleum Production Systems *PTR Pretice Hall*, 1994

[38] Elman, H.C.; Golub, G.H.: Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems *SIAM Journal on Numerical Analysis*, 31 (6), pp 1645-1661, December 1994

[39] Elsharkawy, A.M.; Alikhan, A.A.: Models for Predicting the Viscosity of Middle East Crude Oils *Fuel*, 78, pp 891-903, 1999

[40] Falgout, R. D.; Schroder, J.B.: Non-Galerkin Coarse Grids for Algebraic Multigrid *SIAM Journal on Scientific Computing*, 36(3), pp 309-334, June 2014

[41] Fischer, G.: Lineare Algebra *Vieweg Verlag*, 15th edition, August 2005 (*in German*)

[42] Forsyth, P.A.; Sammon, P.H.: Practical Considerations for Adaptive Implicit Methods in Reservoir Simulations *Journal of Computational Physics*, 62, pp 265-281, 1986

[43] Fung, L.S.K.; Dogru, A.H.: Parallel Unstructured Solver Methods for Complex Giant Reservoir Simulation *SPE Reservoir Simulation Symposium*, February 2007

[44] Gander, M.J,: Schwarz Methods in the Course of Time *Electronic Transactions on Numerical Analysis*, pp 228-255, 2008

[45] Gries, S.; Stüben, K.; Brown, G.L.; Chen, D.; Collins, D.A.: Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations *SPE Journal*, 19 (4), August 2014

[46] Hayder, M.E.; Baddourah, M.A.; Al-Harbi, B.; Al-Zawawi, A.S.; Abouheit, F.F.; Al-Nahdi, U.A.; Al-Zamil, K.S.: Designing a High Performance Computational Platform for Simulation of Giant Reservoir Models *Saudi Aramco Journal of Technology*, pp 56-67, Summer 2013

[47] Heuveline, V.; Lukarski, D.; Weiss, J.-P.: Enhanced Parallel ILU(p)-based Preconditioners for Multi-Core CPUs and GPUs - The Power(q)-Pattern Method *Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)*, 8, 2011

[48] Hysom, D.; Pothen, A.: A Scalable Parallel Algorithm for Incomplete Factor Preconditioning *SIAM Journal on Scientific Computing*, 22 (6), pp 2194-2215, 2001

[49] Isaacson, E.; Keller, H.B.: Analysis of Numerical Methods *Courier Corporation*, 1994

[50] Jaure, S.; Moncorge, A.; de Loubens, R.: Reservoir Simulation Prototyping Platform for High Performance Computing *SPE Large Scale Computing and Big Data Challenges Conference*, September 2014

[51] Janna, C.; Ferronato, M.; Gambolati, G.: Parallel Inexact Constraint Preconditioning for Ill-Conditioned Consolidation Problems *Computational Geosciences*, 16 (3), pp 661-675, June 2012

[52] Jiang, Y.: Techniques for Modeling Complex Reservoirs and Advanced Wells *PhD thesis, Stanford University*, December 2007

[53] Joubert, W.; Oppe, T.; Janardhan, R.; Dearholt, W.: Fully Parallel Global M/ILU Preconditioning for 3-D Structured Problems *The Pennsylvania State University CiteSeerX Archives*

[54] Juliusson, E.; Horne, R.N.: Study and Simulation of Tracer and Thermal Transport in Fractured Reservoirs *35th Workshop on Geothermal Reservoir Engineering, Stanford University*, February 2010

[55] Kim, J.: Sequential Methods for Coupled Geomechanics and Multiphase Simulation *PhD thesis, Stanford University*, February 2010

[56] Kim, J.; Tchelepi, H.A.; Juanes, R.: Stability, Accuracy and Efficiency of Sequential Methods for Coupled Flow and Geomechanics *SPE Journal*, pp 249-262, June 2012

[57] Kim, J.; Tchelepi, H.A.; Juanes, R.: Rigorous Coupling of Geomechanics and Multiphase Flow with Strong Capillarity *SPE Journal*, pp 1123-1139, December 2013

[58] Klausen, R.A.; Winther, R.: Convergence of Multi Point Flux Approximations on Quadrilateral Grids *Numerical Methods for Partial Differential Equations*, 22 (6), pp 1438-1454, November 2006

[59] Klie, H.; Ramé, M.; Wheeler, M.F.: Two-Stage Preconditioning for Inexact Newton Methods in Multiphase Reservoir Simulations *Technical Report, Rice University*, CRPC-TR96641-S, January 1996

[60] Klie, H.; Wheeler, M.F.; Clees, T.; Stüben, K.: Deflation AMG Solvers for Highly Ill-Conditioned Reservoir Simulation Problems *SPE Reservoir Simulation Symposium*, February 2007

[61] Kwok, W.H.F.: Scalable Linear and Nonlinear Algorithms for Multiphase Flow in Porous Media *PhD thesis, Stanford University*, December 2007

[62] Lacroix, S.; Vassilevski, Y.V.; Wheeler, J.: Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly *SIAM Journal on Scientific Computing*, 25, pp 905-926, 2003

[63] Lacroix, S.; Vassilevski, Y.V.; Wheeler, M.F.: Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS), I: Single Processor Case *Numerical Linear Algebra with Applications*, 2000

[64] Li,G.; Wallis, J.; Shaw, G.: A Parallel Linear Solver Algorithm for Solving Difficult Large Scale Thermal Models *SPE Reservoir Simulation Symposium*, SPE-173207, February 2015

[65] Livescu, S.; Durlofsky, L.J.; Aziz, K.; Ginestra, J.C.: A new Model for Simulating Nonisothermal Multiphase Flow in Wellbores and Pipes *Elsevier Science* 2008

[66] Luby, M.: A Simple Parallel Algorithm for the Maximal Independent Set Problem *SIAM Journal of Computing*, 15 (4), pp 1036-1053, November 1986

[67] Ma, S.; Saad, Y.: Distributed ILU(0) and SOR Preconditioners for Unstructured Sparse Linear Systems *SIAM Journal on Computing*, 1998

[68] Manteuffel, T.A.: An Incomplete Factorization Technique for Positive Definite Linear Systems *Mathematics of Computation*, 34 (150), pp 473-797, April 1980

[69] Marsden, J.; Hughes, T.J.R.: Mathematical Foundations of Elasticity *Dover Publications*, 1983

[70] Matthäi, S.K.; Geiger, S.; Roberts, S.G.; Paluszny, A.; Belayneh, M.; Burri, A.; Mezentsev, A.; Lu, H.; Coumou, D.; Driesner, T.; Heinrich, C.A.: Numerical Simulation of Multi-Phase Fluid Flow in Structurally Complex Reservoirs in: Jolley, S.J.; Barr, D.; Walsh, J.J.; Knipe, R.J. (eds): *Struturally Complex Reservoirs.*, Geological Society, London, 405-429, 2007

[71] Meijerink, J.A. and van der Vorst, H.A.: An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-Matrix *Mathematics of Computation*, 31 (137), pp148-162, January 1977

[72] Metsch, B.: Algebraic Multigrid (AMG) for Saddle Point Problems *PhD thesis, University of Bonn*, March 2015

[73] Moncorgé, A.; Tchelepi, H.A.: Stability Criteria for Thermal Adaptive Implicit Compositional Flows *SPE Journal*, pp 311-322, June 2009

[74] Moog, G.: Advanced Discretization Methods for Flow Simulations using Unstructured Grids *PhD thesis, Stanford University*, June 2013

[75] Naji, H.S.: Conventional and Rapid Flash Calculations for the Soave-Redlich-Kwong and Peng-Robinson Equation of State *Emirates Journal for Engineering Research*, 13 (3), pp 81-91, 2008

[76] Naumov, M.: Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU *Nvidia Technical Report*, 001, 2011

[77] Napov, A.; Notay, Y.: An Algebraic Multigrid Method with Guaranteed Convergence Rate *SIAM Journal on Scientific Computing*, 34, pp 1079-1109, 2012

[78] van Odyck, D.E.A.; Bell, J.B.; Monmont, F.; Nikiforakis, N.: The Mathematical Structure of Multiphase Thermal Models of Flow in Porous Media *The Royal Society*, 465 (2102), February 2009

[79] Oertel, K.-D.; Stüben, K.: Multigrid with ILU-Smoothing: Systematic Tests and Improvements *Notes on Numerical Fluid Mechanics*, 23, pp 188-199, 1989

[80] OpenMP comitee: OpenMP Application Program Interface *available at http://www.openmp.org*, Version 3.0, May 2008

[81] Peaceman, D.W.: Interpretation of Well-Block Pressures in Numerical Reservoir Simulation *SPE Journal*, pp 183-194, June 1978

[82] Philip, Z.G.; Jennings Jr., J.W.; Olson, J.E.; Laubach, S.E.; Holder, J.: Modeling Coupled Fracture-Matrix Fluid Flow in Geomechanically Simulated Fracture Networks *SPE Reservoir Evaluation & Engineering*, pp 300-309, August 2005

[83] Plemmons, R.J.: M-Matrix Characterizations. I - Nonsingular M-Matrices *Linear Algebra and its Applications*, 18, pp 175-188, 1977

[84] Ruge, J.: *Private Communication*

[85] Ruge, J.W.; Stüben, K.: Algebraic Multigrid (AMG) *Multigrid Methods, SIAM Frontiers in Applied Mathematics*, 5, Philadelphia, 1986

[86] Rubin, B.; Buchanan, W.L.: A General Purpose Thermal Model *SPE Journal*, pp. 202-214, 1985

[87] Saad, Y.: ILUT: A Dual Threshold Incomplete LU Factorization *Numerical Linear Algebra with Applications*, 1 (4), pp 387-402, August 1994

[88] Saad, Y.: Iterative Methods for Sparse Linear Systems *PWS Publishing*, 1996

[89] Sarma, P.: New Transfer Functions for Simulation of Naturally Fractured Reservoirs with Dual Porosity Models *MSc thesis, Stanford University*, May 2003

[90] Scheichl, R.; Masson R.; Wendebourg, J.: Decoupling and Block Preconditioning for Sedimentary Basin Simulations *Computational Geosciences*, 7 (4), 295-318, 2003

[91] Schiozer, D.J.: Simultaneous Simulation of Reservoir and Surface Facilities *PhD thesis, Stanford University*, March 1994

[92] Settari, A.; Walters, D.A.: Advances in Coupled Geomechanical and Reservoir Modeling with Applications to Reservoir Compaction *SPE Journal*, pp 334-342, September 2001

[93] Silvester, J.R: Determinants of Block Matrices *The Mathematical Gazette*, 84 (501), pp 460-467, November 2007

[94] Sorbie, K.S.; Ryazanov, A.V.; van Dijke, M.I.J: The Structure of Residual Oil as a Function of Wettability Alteration using Pore-Scale Network Modelling *Symposium of the Society of Core Analysts SCA*, Austin TX, September 2011

[95] Stüben, Klaus: An Introduction to Algebraic Multigrid *in [97]*, pp 413-532

[96] Stüben, Klaus: SAMG User's Manual *Fraunhofer SCAI*, continuously updated

[97] Trottenberg, U.; Oosterlee, C.; Schüller, A.: Multigrid *Academic Press*, 2001

[98] van der Vorst, H.: High Performance Parallel Preconditioning *SIAM Journal on Scientific and Statistical Computing*, 10, pp 1174-1185, 1989

[99] Vanek, P.; Mandel, J.; Brezina, M.: Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems *Computing*, 56, pp 179-196; 1996

[100] Varga, R.S.: Matrix Iterative Analysis *Springer*, 2009

[101] Wallis, J.: Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration *SPE Reservoir Simulation Symposium*, 1983

[102] Wallis, J.; Kendall, R.; Little, T.; Nolen, J.: Constrained Residual Acceleration of Conjugate Gradient Acceleration *SPE Reservoir Simulation Symposium*, 1985

[103] Wan, J.: Stabilized Finite Element Methods for Coupled Geomechanics and Multiphase Flow *PhD thesis, Stanford University*, November 2002

[104] Wang, Y.; Hajibeygi, H.; Tchelepi, H.A.: Algebraic Multiscale Solver for Flow in Heterogeneous Porous Media *Journal of Computational Physics*, 259, pp 284-303, February 2014

[105] Wesseling, P.: An Introduction to Multigrid Methods *R.T. Edwards Inc.*, 2004

[106] Whitaker, S.: Flow in Porous Media I: A Theoretical Derivation of Darcy's Law *Transport in Porous Media*, 1 (1), pp 3-25, 1986

[107] Wienands, R.; Joppich, W.: Practical Fourier Analysis for Multigrid Methods *Chapman & Hall/CRC Press*, Boca Raton, Fl, 2005

[108] Wittum, G.: On the Robustness of ILU Smoothing *SIAM Journal on Scientific Statistical Computing*, 10 (4), pp 699-717, 1989

[109] Wong, T.W.:; Firoozabadi, A.; Aziz, K.: Relationship of the Volume-Balance Method of Compositional Simulation to the Newton-Raphson Method *SPE Reservoir Engineering*, pp 415-422, August 1990

[110] Zhou, Y.: Parallel General-Purpose Reservoir Simulation with Coupled Reservoir Models and Multisegment Wells *PhD thesis, Stanford University*, November 2012

[111] Zima, Mirosława: A theorem on the spectral radius of the sum of two operators and its application *Bulletin of the Australian Mathematical Society*, 48 (3), pp 427-434, December 1993

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegeben Teilpublikationen - noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Ulrich Trottenberg betreut worden.

Teilpublikationen:

- Gries, S.; Stüben, K.; Brown, G.L.; Chen, D.; Collins, D.A.: "Advanced AMG Application for Solving CPR-Type Pressure Systems", *SPE Conference on Mathematical Methods in Fluid Dynamics and Simulation of Giant Oil and Gas Reservoirs*, Istanbul, September 2012 (nur Vortrag)
- Gries, S.; Stüben, K.; Brown, G.L.; Chen, D.; Collins, D.A.: "Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations", *SPE Reservoir Simulation Symposium*, The Woodlands (Texas), Februar 2013
- Gries, S.; Stüben, K.; Brown, G.L.; Chen, D.; Collins, D.A.: "Preconditioning for Efficiently Applying Algebraic Multigrid in Fully Implicit Reservoir Simulations", *SPE Journal*, August 2014
- Gries, S.: "Tuning Systems Algebraic Multigrid in Reservoir Simulation", *SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference*, Istanbul, September 2014 (nur Vortrag)
- Gries, S.; Plum, H.-J.: "Status of System-AMG for Reservoir Simulation Applications", *SPE Reservoir Simulation Symposium*, Houston (Texas), Februar 2015

Sankt Augustin, den 09.11.2015
Ort, Datum

Sebastian Gries