Preconditioning for Thermal Reservoir Simulation

Thomas Roy

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2019

À Gabrielle

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Andy Wathen for his guidance and support in the last three years. Without his encouragement and insight, this thesis would not have come to fruition. I would also like to thank my industrial supervisors, Tom Jönsthövel and Chris Lemon, who have gone out of their way to make this project happen. Their expertise in reservoir simulation has shaped my understanding of the field.

My computational tests would not have been possible without the help of the Firedrake developers. For this, I need to thank Lawrence Mitchell, Patrick Farrell, and especially Florian Wechsung for his patience in helping resolve many bugs in my code.

For enabling this DPhil, I deeply appreciate the tireless work of Chris Breward and Colin Please as well as the others who were instrumental to InFoMM. The CDT provided me with an enriching experience and a warm welcome to Oxford that led to many friendships within the cohorts.

I am thankful for my office mates in the S2.33 Matlab Fanbase, Abi, Bogdan, Julien, and Simon, for the many conversations, both mathematical or otherwise. I extend these thanks to the others who joined us on our (too?) many lunch and coffee breaks. For keeping me sane through my time in Oxford, I thank my friends, notably the D&D crew, the lovely Lake Street family, and those in London. I want to specifically thank Kieran who was a constant presence throughout my time here.

I am truly grateful for my parents who have supported me through my many years of education, and my sister whose hard work has been an inspiration.

Finally, I thank my wife Gabrielle who has been by my side through all of it. My motivation stems solely from her, and her faith in me has kept me going for all these years. Hopefully, her long term investment will pay off soon.

Abstract

Multiphase flow through porous media can be modelled as a complex system of partial differential equations. Such models can be used to optimize the recovery of oil and gas from subsurface reservoirs. In the case of highly viscous oils, thermal recovery techniques are typically used to enhance their extraction. To simulate this, models describing the flow of fluids (typically oil, water, and gas) are coupled with a model for heat flow. Thermal reservoir simulation entails solving these highly coupled systems. Their complexity and the computational effort needed to solve them motivate the need for highly efficient solvers.

In reservoir simulation, most of the computational time is spent on solving linearized systems with a preconditioned Krylov subspace iterative method. Industry-standard preconditioning techniques are based on the approach introduced by Wallis in 1983, the Constrained Pressure Residual method (CPR). This preconditioner is a two-stage process involving the solution of a restricted pressure system.

While initially designed for isothermal reservoir simulation, CPR is also the standard for thermal cases. However, its treatment of the conservation of energy equation does not incorporate heat diffusion, which is often dominant in thermal cases. We are interested in preconditioners specifically designed for thermal reservoir simulation. In this thesis, we present an extension of CPR: the Constrained Pressure-Temperature Residual (CPTR) method, where a restricted pressure-temperature system is solved in the first stage. To study the effects of both pressure and temperature on fluid and heat flow, we first consider a model of non-isothermal single-phase flow through porous media. For this model, we develop a block preconditioner with an efficient Schur complement approximation. Then, we extend this method for multiphase flow as a solver for the first stage of CPTR. We present a comparison of the algorithmic performance of the different preconditioning approaches under mesh refinement and parallelization.

Contents

List of Figures

Chapter 1

Introduction

Models of fluid flow in porous media are used in the simulation of applications such as petroleum reservoirs, carbon storage, hydrogeology, and geothermal energy. In some cases, fluid flow must be coupled with heat flow to capture thermal effects. For example, heat from the Earth's core (geothermal energy) flows upward, heating water reservoirs in the crust [13]. These reservoirs can be tapped to generate electricity or heat buildings. This process involves the flow of water, steam, and heat through porous rock.

Oil and gas recovery from petroleum reservoirs is the main application of interest for this thesis. A petroleum reservoir is a subsurface pool where hydrocarbons and water sit deep underground in porous rock trapped under a layer of impermeable rock. This is illustrated in Figure 1.1. The drilling of wells creates a pressure gradient that causes the fluids to flow up to the surface, allowing for the extraction of hydrocarbons. This is known as the Primary Oil Recovery. Once they stop naturally lifting to the surface, water can be injected inside the reservoir to displace some of the remaining hydrocarbons and maintain the pressure difference.

To increase recovery, "Enhanced Oil Recovery" (EOR) techniques are then used. This is especially important for heavy viscous oils. EOR techniques include CO_2 injection, chemical injection, and thermal recovery. The recovery of heavy oils can be facilitated by reducing their viscosity. A variety of thermal recovery techniques [19] are used to achieve that goal.

Steam Injection Cyclic Steam Stimulation is a thermal recovery technique consisting of three stages: injection of steam to increase temperature, a resting period to let the heat diffuse, followed by oil production from that same well. Other techniques — Steam or Hot Water Flooding — use steam or hot water injection wells in conjunction with production wells. These methods combine the viscosity-decreasing effects of

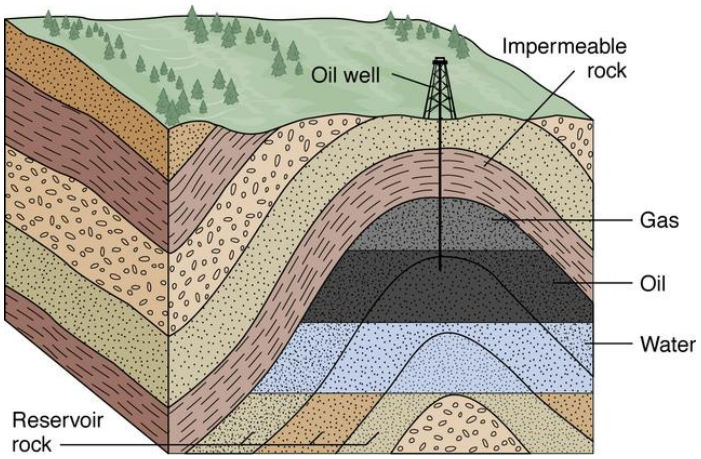


Figure 1.1: A typical oil and gas reservoir (taken from [65]).

heat with the physical displacement of oil being pushed by the water. Steam-Assisted Gravity Drainage (SAGD) is an example of Steam Flooding where a horizontal injection well is drilled above a production well allowing gravity to help with the oil displacement. This process is illustrated in Figure 1.2.

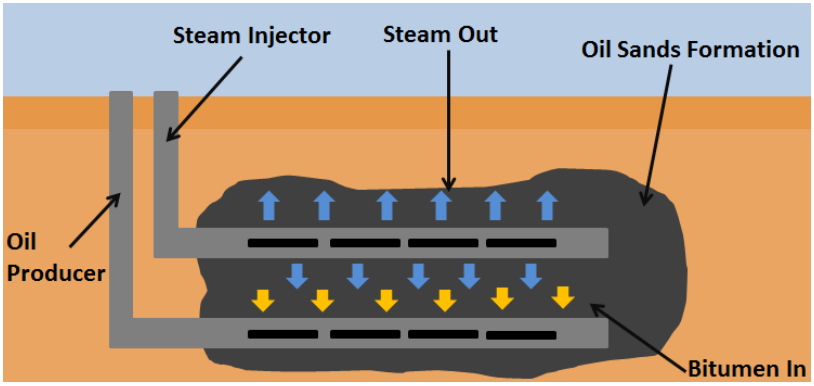


Figure 1.2: Steam-assisted gravity drainage (SAGD) (taken from [34]).

In-Situ Combustion is a thermal recovery technique in which heat is generated through the combustion of the hydrocarbons inside a reservoir by injecting air or oxygen. While less stable, this process is generally cheaper than steam injection. When In-Situ Combustion is successfully applied, distillation and cracking of hydrocarbons can result in the production of lighter and more valuable oil.

A less standard thermal recovery technique is Electromagnetic Heating [78, 89]. An electrical current can be used to increase the temperature of the reservoir, which can be combined with steam injection techniques. Various frequencies are being considered; from low frequency electric resistive heating to high-frequency microwave

heating. Such methods could minimize heat losses that occur during steam injection [63].

Reservoir simulation [7, 74] is the use of mathematical models to predict the flow of fluids through porous media. Petroleum engineers use reservoir simulation to optimize oil recovery processes. For example, history matching uses observed behaviour from the actual reservoir to update the reservoir model [71]. This type of inverse problem, as well as other predictive techniques, require the rapid solution of the forward model. This explains the need for fast and robust commercial reservoir simulators. The simulation of geothermal energy problems is also very similar to petroleum reservoir simulation [37].

The mathematical models solved in reservoir simulators consist of complex systems of partial differential equations (PDEs) describing flow in porous media. The spatial discretization of the PDEs is often done via Finite Volume methods, and the temporal discretization via implicit methods. A Newton-Raphson type method is applied to the resulting nonlinear system, which in turn results in large systems of linearized equations to be solved at each time-step. In commercial reservoir simulators, most of the computational time is spent on solving the resulting linearized systems with a preconditioned Krylov subspace iterative method [86].

In isothermal models, a pressure variable couples a number of secondary variables (saturations/concentrations) which characterize the location of different phases and hydrocarbons. The resulting PDE system is essentially elliptic with respect to pressure and hyperbolic with respect to the secondary variables. Since pressure drives the flow, successful solution techniques usually include a specific treatment of the pressure variable. A classical way to do so is with the Implicit Pressure Explicit Saturation (IMPES) method [91, 95]. The basic idea is to separate the coupled system into a pressure equation and saturation equations which are solved using implicit and explicit time-stepping schemes, respectively. While easy to implement and computationally affordable (only a pressure system is solved implicitly), the stability of the explicit saturation equations requires small time-steps. Thus, IMPES is not efficient for problems with stronger nonlinearities and more than two phases [22].

A more robust approach is to solve all coupled equations simultaneously and implicitly, often referred to as the Fully Implicit Method (FIM). The unconditional stability of FIM implies that very large time-steps can be used. Solving the coupled equations involves solving large and complex linearized systems via iterative methods. Efficient preconditioners are needed to achieve a rapid convergence of those methods. A preconditioner is a linear operator meant to accelerate the convergence of iterative

methods. However, the difficulty of designing efficient preconditioners increases as the system's complexity does. Challenges arise when considering compositional and thermal flow, poromechanics, and other physics of interest.

To avoid having to solve the fully coupled linearized system, a variety of sequential methods exist. These still solve all equations implicitly, but in a sequential manner [60]. Constraints are often applied to the variable set not being solved to improve the convergence of this fixed-point iteration. Sequential implicit methods have been used in commercial reservoir simulation [2, 94, 109]. However, fully implicit methods are often preferred due to their stability. Recent efforts have used Newton's method on the fixed-point iteration to gain quadratic convergence over the usual linear convergence [113]. A careful understanding of the constraints is required for the convergence of these new methods.

Adaptive implicit methods are also used for reservoir simulation [100]. The idea is to find an efficient middle ground between a fully implicit method and IMPES or a sequential implicit method. At a given time-step, different parts of the domain will be solved fully implicitly, while others will use the less coupled methods. The challenge with adaptive implicit methods is to design an efficient switching criterion to determine when the secondary unknowns are solved implicitly or not.

In this thesis, we only consider fully implicit formulations and focus on the required preconditioning strategies.

The industry-standard preconditioner is the Constrained Pressure Residual method (CPR) [105, 106]. It is a two-stage process where a restricted pressure system is solved, followed by an approximate solution of the coupled system. This method was later improved in [54] with the use of Algebraic Multigrid (AMG) [85] as a solver for the pressure subsystem. The second stage is usually an incomplete factorization method such as incomplete LU factorization (ILU) [66].

Multigrid methods [14, 45] combine successively coarser grids and a basic iterative method (relaxation) to reduce all error components. This strategy is "global" and thus effective for elliptic differential operators. Instead of needing a coarse grid hierarchy, AMG constructs the coarse spaces solely with information from the system matrix. This construction relies on heuristics that typically require matrix properties that result from the low-order discretization of an elliptic operator. The elliptic-like nature of the pressure system solved in the first stage of CPR makes it an ideal candidate for the use of multigrid methods. AMG typically requires less work from the user than geometric variants (for example, a mesh hierarchy is not required for AMG). Therefore, it is often preferred in industrial code [20, 44, 49, 72]. Alternatively, an

incomplete factorization method specifically designed for reservoir simulation, Nested Factorization [6], is sometimes used in the first stage of CPR.

The pressure system in the first stage of CPR is approximately decoupled from the other unknowns using a matrix transformation. To date, only minor improvements have been proposed to the decoupling operators used in the original CPR. The alternate block factorization (ABF) decoupling operator, introduced in the general context of systems of PDEs [10], was applied to reservoir simulation in [53]. In [54, 55], the authors introduced two now widely-used decoupling operators, Quasi-IMPES (QI) and True-IMPES (TI), and a less popular method involving local QR decompositions. While QI is purely algebraic in nature, TI can be justified using the PDE structure of the problem. Other approaches include summing the different mass equations [90], and reducing the coupling via least-squares [3]. As an alternative to the algebraic decoupling operators, analytical decoupling techniques have been proposed in [76]. However, it is unclear if this method has any practical application in commercial reservoir simulators.

The use of decoupling operators has been brought into question as they may affect the properties of the pressure system that are required for the fast convergence of AMG within CPR [44, 99]. Even if the original pressure block has fully elliptic properties, the approximately decoupled pressure block may not be amenable for the application of AMG.

Instead of using AMG as an inner solver for the pressure system within a two-stage preconditioner, it has been proposed that it be used for the whole system [27, 44, 99]. Rather than using classical AMG, which is meant for scalar PDEs, this approach uses a version of AMG meant for systems of PDEs [26]. Some versions of this method have a similar philosophy to CPR. Indeed, AMG is applied to the whole system, but all non-pressure variables remain on the fine level. To construct a pressure subsystem which is amenable to the application of AMG, the Dynamic Row-Sum transformation [44] is used as an alternative to previous decoupling operators.

In non-isothermal models, an energy conservation equation is added to the system along with a temperature (or enthalpy) variable. For fully implicit formulations, the industry-standard preconditioner is CPR where temperature variables are grouped with the secondary variables. This is often appropriate since heat is being transported by flow similarly to the saturations. However, heat is also diffused through rock and fluids. Diffusion can dominate in cases where the fluid flow is slow, for instance before viscous oils are properly heated, but also numerically due to mesh refinements. In those cases, the second stage of CPR struggles to capture the heat diffusion, and so

incomplete factorizations with additional fill are needed. Of course, this remedy is not ideal in terms of scalability and memory requirements. A recent example of AMG struggling in the first stage of CPR for thermal simulations was reported in [58]. Accordingly, the authors propose an alternative to AMG for the pressure problem based on block diagonal scaling and matrix equilibration: the SWIFT preconditioner. Another alternative to AMG is proposed in [38], where the matrix structure of the pressure block is used to accelerate the convergence rate of incomplete factorization methods.

How to precondition linear systems resulting from thermal cases is still an open question. Efforts to answer this question are limited in the reservoir engineering community. For sequential implicit methods, the framework developed in [101] has been generalized in [104] for thermal models. Although most recent work about sequential implicit methods relates to coupling poromechanics, there is potential for the thermal problem [113].

For fully implicit methods, one recent contribution is Enhanced CPR (ECPR) [57], which dynamically constructs the system solved in the first stage of CPR allowing additional unknowns than pressure to be included. ECPR constructs this "strong" subsystem by looking at the strength of connections in the system matrix. ECPR suffers from the non-applicability of AMG for the first stage system because its properties are unpredictable. The system AMG preconditioner has also been extended to the thermal case, allowing both pressure and temperature to be considered for the coarse grid hierarchy [42]. However, the community's recent focus is again in coupling fluid flow and poromechanics, rather than thermal flow [43].

For the coupling of single-phase flow with poromechanics, a block preconditioner with an efficient Schur complement approximation was introduced in [112] for a pressure-displacement system, and in [21] for a three-field formulation (where velocity is a variable). A CPR-like two-stage preconditioner allows the extension of the block preconditioner to the multiphase case in [111].

In this thesis, we present an extension of CPR for non-isothermal flow. Instead of solving a restricted pressure system in the first stage of CPR, we solve a restricted pressure-temperature system, resulting in a Constrained Pressure-Temperature Residual method (CPTR). The choice of the solver for the first stage of CPTR is a major challenge for this approach. To investigate pressure-temperature solvers, we first consider single-phase non-isothermal flow. The single-phase model results in a pressure-temperature system of similar properties to the one considered in the first stage of CPTR. On its own, the single-phase case is relevant for simple geothermal energy

and reservoir simulation examples but is also similar to miscible displacement problems [12] (where a concentration plays a similar role to temperature, and molecular diffusion is analogous to heat diffusion) .

For the single-phase case, we present a Schur complement approximation for the pressure-temperature system. Such an approximation leads to an effective block preconditioner. Then, we propose an extension of that method to the multiphase flow situation and use it for the pressure-temperature subsystem in the first stage of CPTR. As an alternative, we also consider applying an unknown-based AMG method to the pressure-temperature subsystem.

In Chapter 2, we describe the mathematical models for non-isothermal flow in porous media for both the single- and multi-phase cases. In Chapter 3, we describe the numerical solution and preconditioning techniques that are standard in reservoir simulation. This is followed by a discussion on the limitations of decoupling operators. A weak formulation of the Finite Volume method is presented in Chapter 4 for simple problems and then applied to the models of Chapter 2. This weak formulation allows the use of the Finite Element library Firedrake [77] and its interface with the solver library PETSc [9]. In Chapter 5, we describe preconditioning approaches for the single-phase case and compare their algorithmic performance. We then extend these preconditioners to the multiphase case in Chapter 6, and present numerical results for two-phase cases. We conclude in Chapter 7 with a discussion on the future direction of this research.

Acknowledgment of funding

This thesis is based on work partially supported by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with Schlumberger.

Chapter 2

Governing Equations

In this chapter, we describe systems of coupled partial differential equations for single- and multi-phase non-isothermal flow in porous media. In each case, the conservation of mass equations could also be used for isothermal models.

2.1 Single-phase flow

Here we describe the equations for single-phase flow in porous media coupled with thermal effects.

2.1.1 Conservation of mass

We begin with the continuity equation which states that the rate at which mass enters the system is equal to the rate of mass which leaves the system plus the accumulation of mass within the system. Additionally, we include a source/sink term which accounts for mass which is added or removed from the system. We have

$$\phi \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = f \quad \text{in } \mathbb{R}_+ \times \Omega, \tag{2.1}$$

where ϕ is the porosity field of the rock, ρ is the density of the fluid, $\mathbf{u}$ is the fluid velocity, f is a source/sink term, and Ω is the spatial domain in $\mathbb{R}^d$, $d = 2, 3$. The source/sink term f represents injection/production wells and is given in Section 2.4. We consider a fixed rock strata so that the porosity field ϕ is constant in time.

In fluid dynamics models such as the Navier-Stokes equations, the velocity $\mathbf{u}$ is commonly defined using a momentum equation. For flow in porous media, this may only be necessary on the pore-scale. On the macroscale, flow in porous media can be modelled by Darcy's Law [29]. Although it was originally determined experimentally

by Darcy, it has since been derived from the Navier-Stokes equations via homogenization [110].

We assume that the velocity follows Darcy's law, i.e.

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g}), \tag{2.2}$$

where p is the pressure, $\mathbf{K}$ is the permeability tensor field, μ is the viscosity, and $\mathbf{g}$ is the gravitational acceleration vector. The density and viscosity are functions of pressure and temperature given in Section 2.5. Explicitly substituting (2.2) in (2.1), we get

$$\phi \frac{\partial \rho}{\partial t} - \nabla \cdot \left(\rho \frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g}) \right) = f \quad \text{in } \mathbb{R}_+ \times \Omega. \tag{2.3}$$

We also assume Neumann and Dirichlet boundary conditions

$$-\frac{\mathbf{K}}{\mu}(\nabla p - \rho \mathbf{g}) \cdot \mathbf{n} = g_N \text{ on } \Gamma_N, \quad \text{and} \quad p = g_D \text{ on } \Gamma_D, \tag{2.4}$$

where g_N is Neumann boundary data, g_D is Dirichlet boundary data, $\mathbf{n}$ is the unit outward normal vector on $\partial\Omega = \Gamma_N \cup \Gamma_D$, and $\Gamma_D \cap \Gamma_N = \varnothing$.

2.1.2 Conservation of energy

Similarly, we have a conservation of energy equation for heat energy. Note that formulations where enthalpy is an independent variable are common, but here we consider temperature as an independent variable as in a reference commercial reservoir simulator [30]. Heat energy is not only being transported by fluid flow, but also conducted through rock and fluid . We get the following advection-diffusion equation:

$$\phi \frac{\partial}{\partial t}(\rho U) + (1-\phi)\frac{\partial}{\partial t}(\rho_r H_r) + \nabla \cdot (\rho H \mathbf{u}) + \nabla \cdot \mathbf{q} = f_T \quad \text{in } \mathbb{R}_+ \times \Omega, \tag{2.5}$$

where U is the internal energy of the fluid, ρ_r and H_r are the density and enthalpy of the rock, respectively, H is the enthalpy of the fluid, $\mathbf{q}$ is the heat flux, and f_T is a source/sink term representing wells or heaters, for which details are given in Section 2.4. In the case of gases, the internal energy can be described by the relation

$$U = H - \frac{p}{\rho}, \tag{2.6}$$

and in the case of liquids and solids by $U = H$. We explicitly replace the enthalpies by simple approximations, the following linear relations with respect to the temperature T:

$$H = c_v T, \qquad H_r = c_r T, \tag{2.7}$$

where c_v and c_r are the specific heat of the fluid and rock, respectively, ρ_r is the density of the rock. Higher-order polynomial correlations for the enthalpy of fluids are also common (see for example [28]). Here, $\rho c_v T$ represents the energy density of the fluid. For the numerical tests in this thesis, we only consider liquids, and so (2.5) becomes

$$\phi\frac{\partial}{\partial t}(\rho c_v T) + (1-\phi)\frac{\partial}{\partial t}(\rho_r c_r T) + \nabla\cdot(\rho c_v T\mathbf{u}) + \nabla\cdot\mathbf{q} = f_T \quad \text{in } \mathbb{R}_+\times\Omega. \tag{2.8}$$

The influence of the pressure on the internal energy (2.6) in the case of gases will not considered in the methods presented in Chapters 5 and 6. Considering this effect may reduce the quality of some of the approximations presented (see in particular Section 5.1.3.3.

Furthermore, we assume that the heat flux follows Fourier's law, i.e.

$$\mathbf{q} = -k_T\nabla T, \tag{2.9}$$

where k_T is the thermal conductivity field. We choose the following simple relation:

$$k_T = \phi k_{T,r} + (1-\phi)k_{T,f}, \tag{2.10}$$

where $k_{T,r}$ and $k_{T,f}$ are the conductivities of the rock and the fluid, respectively. Substituting Fourier's law into (2.8), we get

$$\phi\frac{\partial}{\partial t}(\rho c_v T) + (1-\phi)\frac{\partial}{\partial t}(\rho_r c_r T) + \nabla\cdot(\rho c_v T\mathbf{u}) - \nabla\cdot(k_T\nabla T) = f_T \quad \text{in } \mathbb{R}_+\times\Omega. \tag{2.11}$$

Assuming Darcy's law, we obtain

$$\phi\frac{\partial}{\partial t}(\rho c_v T) + (1-\phi)\frac{\partial}{\partial t}(\rho_r c_r T) - \nabla\cdot\left(\rho c_v T\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g})\right) - \nabla\cdot(k_T\nabla T) = f_T \quad \text{in } \mathbb{R}_+\times\Omega. \tag{2.12}$$

We also assume Neumann and Dirichlet boundary conditions

$$-\left(\rho c_v T\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g}) + k_T\nabla T\right)\cdot\mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \quad \text{and} \quad T = g_D^T \text{ on } \Gamma_D^T, \tag{2.13}$$

where g_N^T is Neumann boundary data, g_D^T is Dirichlet boundary data, $\partial\Omega = \Gamma_N^T \cup \Gamma_D^T$, and $\Gamma_D^T \cap \Gamma_N^T = \varnothing$.

2.1.3 Coupled problem

We assume that ρ and μ are empirically determined functions of pressure and temperature. Our choices, which are representative for heavy oil reservoirs, are given in Section 2.5.

We are interested in solving the boundary value problem for p and T comprising of the equations (2.3), (2.12) together with boundary conditions (2.4) and (2.13), and prescribed initial conditions for p and T.

We will develop new preconditioning strategies for the resulting system for non-isothermal single-phase flow. The discretization and solution techniques for this system are given in Section 4.3 and Chapter 5, respectively.

2.2 Multiphase flow

In multiphase flow, the number of phases is typically two or three, for example, oil, water, and gas.

The pores in a petroleum reservoir can contain both hydrocarbons and water, which is both connate and injected to improve recovery. If the fluids are immiscible, they are often referred to as phases. On the microscale, these do not form a solution, but, on the macroscale, they are assumed to be present at the same location.

The volume fraction occupied by each phase α is called the saturation of that phase, denoted S_α. We have the following saturation constraint:

$$\sum_\alpha S_\alpha = 1. \tag{2.14}$$

In the absence of phase change, saturations only vary when one phase displaces the other through fluid flow. The ability of a phase α to move is represented by the relative permeability $k_{r\alpha}$, a dimensionless function of S_α.

2.2.1 Conservation of mass

For each phase α, we have a conservation of mass equation of the following form:

$$\phi \frac{\partial (S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega, \tag{2.15}$$

where, as before, ϕ is the porosity of the rock, ρ_α is the density of the fluid, $\mathbf{u}_\alpha$ is the fluid velocity, f_α is a source/sink term, and Ω is the spatial domain. We further

assume that the velocity of each phase α follows Darcy's law, which generalizes in the multiphase case [69] to

$$\mathbf{u}_\alpha = -\mathbf{K}\frac{k_{r\alpha}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \tag{2.16}$$

where p_α is the pressure, μ_α is the viscosity, and, as before, $\mathbf{K}$ is the permeability tensor and $\mathbf{g}$ is gravitational acceleration. Due to surface tension, the pressures of the phases differ. The difference between the pressures of the phases α and β

$$p_{c\alpha\beta} = p_\alpha - p_\beta, \qquad \alpha \neq \beta, \tag{2.17}$$

is called the capillary pressure and is usually assumed to be a function of S_α and S_β. A result of surface tension between phases, capillary pressure adds a diffusive effect for the saturations, which may dominate in some applications and models. There are certainly contexts in which it can be important (for example viscous fingering [88]), but in the context of this thesis, we ignore the effects of capillary pressures so that we only have one pressure $p = p_\alpha$ for all phases α. When the diffusive effect of capillary pressures is significant, the equations are parabolic with respect to the saturations, allowing for different solution strategies including multigrid methods for the saturations [17].

Then, (2.1) becomes

$$\phi\frac{\partial(S_\alpha\rho_\alpha)}{\partial t} - \nabla\cdot\left(\rho_\alpha\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha\mathbf{g})\right) = f_\alpha \quad \text{in } \mathbb{R}_+\times\Omega. \tag{2.18}$$

We also assume Neumann and Dirichlet boundary conditions

$$-\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha\mathbf{g})\cdot\mathbf{n} = g_{N,\alpha} \quad \text{on } \Gamma_N, \quad \text{for all } \alpha, \tag{2.19}$$

$$S_\alpha = g_{D,\alpha}, \text{ for all } \alpha, \quad p = g_D \quad \text{on } \Gamma_D, \tag{2.20}$$

where $g_{N,\alpha}$ is Neumann boundary data for the flow of phase α, g_D and $g_{D,\alpha}$ are Dirichlet boundary data for pressure and saturation α, respectively, $\mathbf{n}$ is the unit outward normal vector on $\partial\Omega = \Gamma_N \cup \Gamma_D$, and $\Gamma_D \cap \Gamma_N = \varnothing$.

2.2.2 Conservation of energy

Similarly, we have a conservation of energy equation for heat energy. We get the following advection-diffusion equation:

$$\frac{\partial}{\partial t}\left(\phi\sum_\alpha \rho_\alpha S_\alpha U_\alpha + (1-\phi)\rho_r H_r\right) + \nabla\cdot\sum_\alpha \rho_\alpha H_\alpha \mathbf{u}_\alpha - \nabla\cdot(k_T\nabla T) = f_T \quad \text{in } \mathbb{R}_+\times\Omega, \tag{2.21}$$

where U_α and H_α are the internal energy and enthalpy of phase α, respectively, and, as before, k_T is the thermal conductivity field, H_r is the enthalpy of the porous medium, ρ_r is its density, and f_T is a source/sink term. In the multiphase case, the conductivity field is given by the simple volume-weighted relation

$$k_T = (1-\phi)k_{T,r} + \phi \sum_\alpha S_\alpha k_{T,\alpha}, \tag{2.22}$$

where $k_{T,r}$ and $k_{T,\alpha}$ are thermal conductivities of the rock and the phase α, respectively.

Similarly to the single-phase case, the internal energy can be described in the case of gases by the relation

$$U_\alpha = H_\alpha - \frac{\rho_\alpha}{p_\alpha}, \tag{2.23}$$

and in the case of liquids and solids by $U_\alpha = H_\alpha$. We explicitly replace enthalpies by this simple linear relation with respect to the temperature T

$$H = c_{v,\alpha} T, \qquad H_r = c_r T, \tag{2.24}$$

where $c_{v,\alpha}$ and c_r are the specific heat of the phase α and the rock, respectively. Here, $c_{v,\alpha}T$ represents the enthalpy of the phase α, and $\rho_\alpha c_{v,\alpha}$, its energy density. In the case of liquid phases, (2.21) becomes

$$\frac{\partial}{\partial t}\left(\phi \sum_\alpha \rho_\alpha S_\alpha c_{v,\alpha} T + (1-\phi)\rho_r c_r T\right) + \nabla \cdot \sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha - \nabla \cdot (k_T \nabla T) = f_T \quad \text{in } \mathbb{R}_+ \times \Omega. \tag{2.25}$$

Assuming Darcy's law, we get

$$\frac{\partial}{\partial t}\left(\phi \sum_\alpha \rho_\alpha S_\alpha c_{v,\alpha} T + (1-\phi)\rho_r c_r T\right) - \nabla \cdot \sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) - \nabla \cdot (k_T \nabla T) = f_T \quad \text{in } \mathbb{R}_+ \times \Omega. \tag{2.26}$$

We also assume Neumann and Dirichlet boundary conditions

$$-\left(\sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) + k_T \nabla T\right) \cdot \mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \; T = g_D^T \text{ on } \Gamma_D^T, \tag{2.27}$$

where g_N^T is Neumann boundary data, g_D^T is Dirichlet boundary data, $\partial\Omega = \Gamma_N^T \cup \Gamma_D^T$, and $\Gamma_D^T \cap \Gamma_N^T = \varnothing$.

2.2.3 Coupled problem

We assume that the relative permeability $k_{r\alpha}$, the density ρ_α, and the viscosity μ_α of phase α are empirically determined functions of saturations, pressure, and temperature. Our choices, which are representative of heavy oil reservoir simulation, are given in Section 2.5.

Since the saturations satisfy the constraint (2.14), one of the saturation variables is explicitly replaced

$$S_\beta = 1 - \sum_{\alpha \neq \beta} S_\alpha. \tag{2.28}$$

Thus, if the following system is for the non-isothermal flow of n_{phases} phases, it consists of $n_{\text{phases}} + 1$ equations and unknowns.

We are interested in solving the boundary value problem for p, T, S_α for all $\alpha \neq \beta$ satisfying equations (2.18), (2.26) together with boundary conditions (2.19), (2.20), (2.27), where initial conditions for p, T, and S_α are prescribed, and we explicitly eliminate S_β via (2.28).

In this thesis, we will develop preconditioning strategies for the above system for non-isothermal multiphase flow. The discretization and solution techniques for this system are given in Section 4.4 and Chapter 6, respectively.

2.3 Compositional flow

The multiphase flow model described in Section 2.2 does not allow for mass transfer between phases. A more general system describes the fluids as different components that may reside in multiple phases. This is known as a compositional model. A simple and widely-used example of such a model is the Black-Oil model [101]. In this model, there are three phases: liquid, vapor, and aqua, and three components: oil, water, and gas. In the typical model, oil can either be in the liquid or vapor phase, gas can be in all three phases, and water is in the aqueous phase only.

As part of this thesis, we will not perform numerical experiments for compositional flow. However, since we sometimes comment about the applicability of different methods for the compositional case, we describe a general model here.

The fluid velocity still follows Darcy's Law, but instead of having mass conservation for each phase, we now have mass conservation for each component. For a model

with n_{comp} components and n_{phases} phases, we have mass conservation equations

$$\frac{\partial}{\partial t}\left(\phi \sum_{\alpha=1}^{n_{\text{phases}}} X_{\alpha,i}\rho_\alpha S_\alpha\right) - \nabla \cdot \sum_{\alpha=1}^{n_{\text{phases}}} X_{\alpha,i}\rho_\alpha \mathbf{K}\frac{k_{r\alpha}}{\mu_\alpha}(\nabla p_\alpha - \rho_\alpha \mathbf{g}) = \sum_{\alpha=1}^{n_{\text{phases}}} x_{\alpha,i}\rho_\alpha q_\alpha, \quad i = 1, \ldots, n_{\text{comp}}, \tag{2.29}$$

where $X_{\alpha,i}$ is the mass fraction of component i within phase α. Additional to the saturation constraint (2.14), we have the following constraints for the mass fractions:

$$\sum_{i=1}^{n_{\text{comp}}} X_{\alpha,i} = 1, \qquad \alpha = 1, \ldots, n_{\text{phases}}. \tag{2.30}$$

The system comprising of (2.14), (2.17), (2.29), and (2.30) still has more unknowns than equations and thus requires closure equations. In general, equilibrium relations describing the mass distribution of hydrocarbon components into phases are added to the system. These are given in the following form:

$$f_{\alpha,i}(p_\alpha, X_{\alpha,1}, \ldots, X_{\alpha,n_{\text{comp}}}) = f_{\beta,i}(p_\beta, X_{\beta,1}, \ldots, x_{\beta,n_{\text{comp}}}), \tag{2.31}$$

where $f_{\alpha,i}$ is the fugacity function of the i^{th} component in the α-phase, $i = 1, \ldots, n_{\text{comp}}$, $\alpha, \beta = 1, \ldots, n_{\text{phases}}$, $\alpha \neq \beta$. Often, a simple approach with tabulated K-values is used to describe the equilibrium relations, such that

$$X_{\alpha,i} = K_{\alpha,i}(p_\alpha)X_{\beta,i}, \qquad \alpha, \beta = 1, \ldots, n_{\text{phases}}, \quad \alpha \neq \beta, \quad i = 1, \ldots, n_{\text{comp}}, \tag{2.32}$$

where $K_{\alpha,i}$ is typically determined experimentally.

The compositional model described above could be used on its own as an isothermal model, or coupled with the conservation of energy equation given by (2.21). In the compositional case, the enthalpy of phase α is given by

$$H_\alpha = \sum_{i=1}^{n_{\text{comp}}} X_{\alpha,i}H_{\alpha,i}, \tag{2.33}$$

where $H_{\alpha,i}$ is the partial enthalpy of component i in phase α. Simple approximations of the enthalpies are given by

$$H_{\alpha,i} = c_{v,\alpha,i}T, \tag{2.34}$$

where $c_{v,\alpha,i}$ represents the specific heat of component i in phase α. The internal energy is given by a relation of the form (2.23).

The system comprising of (2.14), (2.17), (2.29)–(2.31) and (2.21) consists of $(n_{\text{comp}} + 2) \times n_{\text{phases}}$ equations and unknowns. As is done in Section 2.2, we can

eliminate one saturation unknown and all but one pressure unknown. Similarly, we can eliminate $(n_{\text{comp}} + 1) \times n_{\text{phases}}$ concentrations by using (2.30) and (2.31). The problem then comprises of a system of n_{comp} mass conservation equations (2.29) with $n_{\text{phases}} - 1$ saturation unknowns, $n_{\text{comp}} - n_{\text{phases}}$ concentration unknowns, and one pressure unknown p, so $n_{\text{unknowns}} = n_{\text{comp}}$. Note that the choice of which saturation and concentrations are eliminated may have an effect on the properties of the linearized systems. We will comment on this in Sections 3.4.3 and 3.4.4.3.

2.4 Sources and sinks

Efficient recovery techniques for heavy oils include some way of reducing their viscosity, for example by increasing the temperature inside the reservoir. As explained in Chapter 1, this is commonly done by injecting hot fluids (usually steam or hot water) or by electromagnetic heating [89].

We first consider source/sink terms representing injection and production wells for the single-phase flow case. A simple way to model these is by using point sources/sinks

$$f(\mathbf{x}) = \sum_i q^i_{\text{inj}}(p,T)\delta(\mathbf{x} - \mathbf{x}^i_{\text{inj}})\rho(p,T_{\text{inj}}) - \sum_j q^j_{\text{prod}}(p,T)\delta(\mathbf{x} - \mathbf{x}^j_{\text{prod}})\rho(p,T), \quad (2.35)$$

$$f_T(\mathbf{x}) = \sum_i q^i_{\text{inj}}(p,T)\delta(\mathbf{x} - \mathbf{x}^i_{\text{inj}})\rho(p,T_{\text{inj}})c_v T_{\text{inj}} - \sum_j q^j_{\text{prod}}(p,T)\delta(\mathbf{x} - \mathbf{x}^j_{\text{prod}})\rho(p,T)c_v T, \quad (2.36)$$

where $\mathbf{x}_{\text{inj}}$ and $\mathbf{x}_{\text{prod}}$ represent the locations of injection and production wells, respectively, $\delta(\mathbf{x})$ is the Dirac delta function, q^i_{inj} and q^j_{prod} are the wells' injection and production rates, respectively.

The production rate q_{prod} is usually given by a constant target production rate. Similarly, the injection rate q_{inj} is given by a target injection rate. These rates can only be maintained if the pressure at the production well does not drop below a minimum pressure, and the pressure at the injection well does not go above a maximum pressure. In those cases, a well model is required. We consider the commonly used Peaceman well model [23, 75] for anisotropic media with $\mathbf{K} = \text{diag}(K_x, K_y, K_z)$ as the permeability tensor field. In this case, the rates are given by

$$q = \frac{2\pi h K_e}{\mu \ln(r_e/r_w)}(p_{bh} - p), \quad (2.37)$$

where h is the height of well opening, $K_e = \sqrt{K_x K_y}$ is the equivalent permeability, p_{bh} is the bottom-hole pressure, r_w is the well radius, and r_e is the equivalent radius

which can be calculated using

$$r_e = \frac{0.14\left((K_y/K_x)^{1/2}D_x^2 + (K_x/K_y)^{1/2}D_y^2\right)^{1/2}}{0.5\left((K_y/K_x)^{1/4} + (K_x/K_y)^{1/4}\right)}, \tag{2.38}$$

where D_x and D_y are the horizontal lengths of the grid cell. Since we want to allow mesh refinements, we do not want the model to change as we vary the grid size. Therefore, we arbitrarily fix $D_x = D_y = 5$ meters, and also choose $h = 5$ meters and $r_w = 0.1$ meters.

Using a well model usually requires solving additional coupled equations for each well. Since we are using point source/sink wells, we do not have to solve any well equations to determine rates in our simplified model.

Oil recovery techniques for heavy oils can include electromagnetic heating. This can be expressed as a source term for the energy equation. For simplicity, we do not use an electromagnetic model and choose the simple function

$$f_T = \sum_i U_{\text{heater}}(p,T)\delta(\mathbf{x} - \mathbf{x}^i_{\text{heater}})(T_{\text{heater}} - T), \tag{2.39}$$

where $\mathbf{x}_{\text{heater}}$ represent the locations of heaters, U_{heater} is the heat transfer coefficient, and T_{heater} is the target heating temperature. For our simulations, we have a heating coefficient of 5.44409×10^{-6} $\text{Js}^{-1}\text{K}^{-1}$. For simplicity, we also choose T_{heater} to be the same as T_{inj}.

We also consider source/sink terms representing injection and production wells for the multiphase flow case. Again, we can model these by using point sources/sinks

$$\begin{aligned} f_\alpha(\mathbf{x}) = &\sum_i q^i_{\alpha,\text{inj}}(p,T,S_\alpha)\delta(\mathbf{x} - \mathbf{x}^i_{\text{inj}})\rho_\alpha(p,T_{\text{inj}}) \\ &- \sum_j q^j_{\alpha,\text{prod}}(p,T,S_\alpha)\delta(\mathbf{x} - \mathbf{x}^j_{\text{prod}})\rho_\alpha(p,T), \end{aligned} \tag{2.40}$$

$$\begin{aligned} f_T(\mathbf{x}) = \sum_\alpha \Bigg(&\sum_i q^i_{\alpha,\text{inj}}(p,T,S_\alpha)\delta(\mathbf{x} - \mathbf{x}^i_{\text{inj}})\rho_\alpha(p,T_{\text{inj}})c_{v,\alpha}T_{\text{inj}} \\ &- \sum_j q^j_{\alpha,\text{prod}}(p,T,S_\alpha)\delta(\mathbf{x} - \mathbf{x}^j_{\text{prod}})\rho_\alpha(p,T)c_{v,\alpha}T \Bigg), \end{aligned} \tag{2.41}$$

where $q^i_{\alpha,\text{inj}}$ and $q^j_{\alpha,\text{prod}}$ are the wells' injection and production rates, respectively, for phase α. The rates can be defined as above.

2.5 Physical quantities

The relative permeabilities $k_{r\alpha}$, the densities ρ_α, and viscosities μ_α are empirically determined functions of saturations, temperature, and pressure. It is standard practice in reservoir simulation to get such quantities by interpolating values from "input tables", but we will instead choose empirically determined correlations. For the numerical examples in this thesis, we will consider the flow of two phases: a heavy oil and water, denoted o and w, respectively. We use oil in the single-phase case and drop the o subscript for simplicity.

The oil density ρ_o and viscosity μ_o are empirically determined functions of temperature and pressure. For the test cases in Chapters 5 and 6, we will use the empirical laws below.

For the viscosity of oil, we choose the following correlation [11]:

$$\mu_o(T_{\mathrm{F}}) = 10^{A_1\gamma_{\mathrm{API}}+A_2} T_{\mathrm{F}}^{A_3\gamma_{\mathrm{API}}+A_4}, \tag{2.42}$$

which takes temperature T_{F} in °F and returns viscosity in cp (0.001 kg m^{-1} s^{-1}). The viscosity as a function of temperature (in Kelvin) is illustrated in Figure 2.1. We see that the viscosity of the heavy oil varies hugely with temperature. The dimensionless parameters A_i can be found in Table 2.1. The American Petroleum Institute (API) gravity γ_{API} is a measure of how heavy or light a petroleum liquid is compared to water: if its API is greater than 10, then it is lighter and floats on water; if less than 10, it is heavier and sinks. We can calculate the API gravity from the specific gravity γ_{SG} (ratio of the density of the petroleum liquid to the density of water, at 60° F) using the following formula:

$$\gamma_{\mathrm{API}} = \frac{141.5}{\gamma_{\mathrm{SG}}} - 131.5. \tag{2.43}$$

For our test cases, we choose an API gravity of 10. For the density of oil, we use the following correlation:

$$\rho_o(p,T) = \rho_0 e^{c(p-p_0)} e^{\beta(T-T_0)}, \tag{2.44}$$

where p_0, T_0 are reference pressure and temperature, respectively, and ρ_0 is the density at those values, c is a compressibility coefficient, and β is a thermal expansion coefficient. Values representative to those used in reservoir simulation are $p_0 = 1.01325$ bar, T_0= 288.7056 K (60 °F), $c = 5.5 \times 10^{-5}$ bar^{-1}, and $\beta = 2.5 \times 10^{-4}$ K^{-1}. Given a specific gravity, we have $\rho_0 = \gamma_{\mathrm{SG}}\ \rho_w$, where $\rho_w = 999$ kg m^{-3} is the density of water at the reference temperature and pressure.

Table 2.1: Parameters for the Bennison viscosity correlation

A_1	A_2	A_3	A_4
-0.8021	23.8765	0.31458	-9.21592

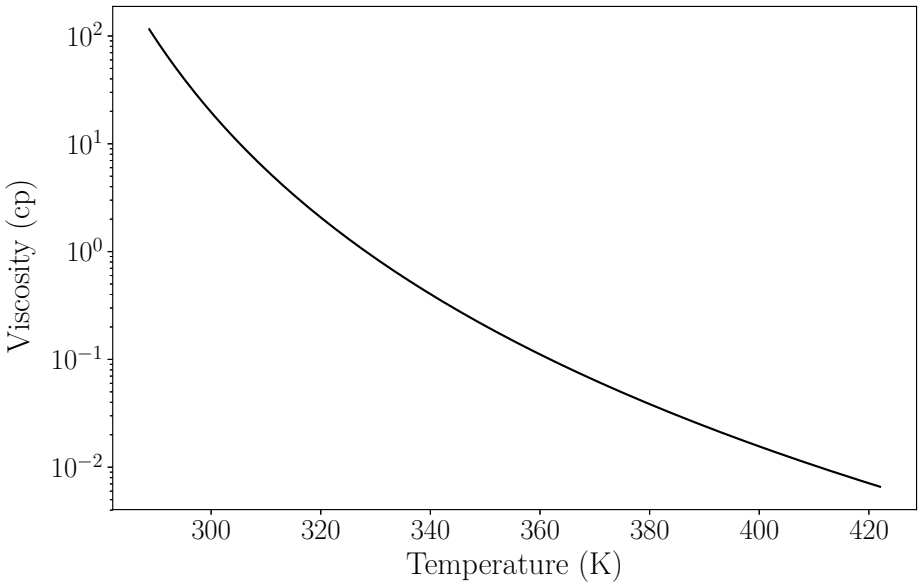


Figure 2.1: The Bennison viscosity correlation for heavy oil.

For the water viscosity μ_w, we use the following correlation [41]:

$$\mu_w(T_{\mathrm{F}}) = \frac{A}{-1 + BT_{\mathrm{F}} + CT_{\mathrm{F}}^2}, \tag{2.45}$$

which takes temperature T_{F} in °F and returns viscosity in cp (0.001 kg m^{-1} s^{-1}). The parameters A, B, C can be found in Table 2.2.

Table 2.2: Parameters for the water viscosity correlation

A	B	C
2.1850	0.04012	5.1547×10^{-6}

For the water density, we use Trangenstein's modification of Kell's formula [51] given by

$$\rho_w(p, T_{\mathrm{C}}) = \frac{E_0 + E_1T + E_2T^2 + E_3T^3 + E_4T^4 + E_5T^5}{1 + E_6T} e^{C_w(p-E_7)}, \tag{2.46}$$

which takes temperature T_{C} in °C, and pressure p in MPa. The parameters are given in Table 2.3.

Table 2.3: Parameters for the water density correlation

E_0	E_1	E_2	E_3	E_4
999.83952	16.955176	-7.987$\times 10^{-3}$	-46.170461$\times 10^{-6}$	105.56302$\times 10^{-9}$

E_5	E_6	E_7	C_w
-280.54353$\times 10^{-12}$	16.87985$\times 10^{-3}$	10.2	3.98854$\times 10^{-4}$

Relative permeability typically reflects that the flow of each phase is inhibited by the presence of others. Thus, we simply set it to be equal to the phase saturation. If we set $S_w = 1 - S_o$, we get $k_{ro} = S_o$, and $k_{rw} = 1 - S_o$.

Chapter 3

Numerical Solution and Current Approaches

The governing equations of Chapter 2 are conservation laws and can be written in the general form

$$\frac{\partial \mathbf{M}}{\partial t} = \nabla \cdot \mathbf{F} + \mathbf{Q}, \tag{3.1}$$

where $\mathbf{M}$ is a vector field describing the mass (or energy) density, $\mathbf{F}$ is a tensor field describing fluid (or thermal) flow, and $\mathbf{Q}$ is a vector field describing source/sink terms such as wells. These are all functions of the solution vector $\mathbf{u}$ which includes pressure and secondary unknowns (such as saturations, concentrations, and temperature). The term unknown will refer to a physical quantity (e.g. pressure) or the discrete variables associated with it. The term variable used alone refers to all discrete variables.

In this chapter, we discuss numerical methods that are standard in commercial reservoir simulators. While these methods were initially developed for isothermal models, they are also used for thermal models. The energy conservation equation and temperature are different from the mass conservation equations and saturations/concentrations. In the preconditioning approach of current reservoir simulators, however, the energy conservation equation and temperature unknown are simply treated in the same manner as the other equations and non-pressure unknowns. We will discuss alternatives in Chapters 5 and 6.

In Section 3.1, we give a brief overview of the discretization and linearization, which result in the linearized systems to be solved with Krylov subspace methods, described in Section 3.2. In Section 3.3, we describe the industry-standard Constrained Pressure Residual preconditioner and its different ingredients. We then discuss in Section 3.4 the properties of the different blocks of our system and their consequences on the applicability of decoupling operators.

3.1 Discretization and linearization

Modern spatial discretization methods for reservoir models need to account for higher complexity reservoir geometries, for which unstructured meshes are used (although Cartesian grids are still commercially relevant). Most commercial reservoir simulators use Finite Volume methods because of their intuitive nature and simple implementation, often preferred by reservoir engineers.

Here, we give a brief description of the spatial discretization via the Finite Volume method for systems of the form (3.1). In Chapter 4, we will detail how to implement a variational formulation of the Finite Volume method for the models used in our numerical tests.

We assume that the spatial domain can be divided into a set of non-overlapping polyhedral discrete volumes in which the properties of the rock and fluids are constant. We apply a Finite Volume discretization to (3.1). By integrating over the grid cell V_i, which is used as a control volume, and using the Divergence Theorem we obtain

$$\int_{V_i} \frac{\partial \mathbf{M}}{\partial t} \, \mathrm{d}V_i = \int_{V_i} (\nabla \cdot \mathbf{F} + \mathbf{Q}) \, \mathrm{d}V_i = \oint_{\partial V_i} \mathbf{F} \cdot \mathbf{n} \; \mathrm{d}S + \int_{V_i} \mathbf{Q} \, \mathrm{d}V_i, \tag{3.2}$$

where ∂V_i is the boundary of the cell V_i and $\mathbf{n}$ is the unit outward normal vector to ∂V_i. Assuming that the solution $\mathbf{u}$ is piecewise constant, the equation simplifies to the quadrature

$$\frac{\partial \mathbf{M}_i}{\partial t} = \frac{1}{|V_i|} \oint_{\partial V_i} \mathbf{F} \cdot \mathbf{n} \; \mathrm{d}S + \mathbf{Q}_i = \frac{1}{|V_i|} \sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} + \mathbf{Q}_i, \tag{3.3}$$

where $\mathbf{M}_i$ is the mass (or energy) in cell i, $\mathcal{N}(i)$ denotes the neighbouring cells to cell i, $|V_i|$ is the volume of cell i, a_{ij} is the area of the facet between cells i and j, and $\mathbf{n}_{ij}$ is its unit outward normal vector from cell i to cell j. $\mathbf{F}_{ij}$ describes the flux between cell i and j. Generally, this is approximated using the two cells adjacent to the given face, which is known as a two-point flux approximation (TPFA). We will use TPFA in Chapter 4.

Applying the implicit Euler's method to (3.3), we obtain the following nonlinear system for the solution vector at time t_n, $\mathbf{u}^n$:

$$\mathbf{\Psi}_i = \frac{\mathbf{M}_i(\mathbf{u}_i^n) - \mathbf{M}_i(\mathbf{u}_i^{n-1})}{\Delta t} - \frac{1}{|V_i|} \sum_{j \in \mathcal{N}(i)} a_{ij} \mathbf{F}_{ij}(\mathbf{u}_i^n, \mathbf{u}_j^n) \cdot \mathbf{n}_{ij} - \mathbf{Q}_i(\mathbf{u}_i^n) = 0 \text{ for all } i, \tag{3.4}$$

where $\mathbf{u}_i^n$ is the solution at cell i and time t_n, and $\mathbf{u}^{n-1}$ is the known solution at time t_{n-1}.

Applying the Newton-Raphson method, we obtain a linear system of the form

$$M_{ii}\delta\mathbf{u_i} - \sum_{j\in\mathcal{N}(i)} F_{ji}\delta\mathbf{u_i} + \sum_{j\in\mathcal{N}(i)} F_{ij}\delta\mathbf{u_j} = -\mathbf{\Psi}_i, \tag{3.5}$$

where

$$M_{ii} = \frac{\partial}{\partial \mathbf{u}_i^n}\left(\frac{\mathbf{M}_i(\mathbf{u}_i^n)}{\Delta t} - \mathbf{Q}_i(\mathbf{u}_i^n)\right), \quad \text{and} \quad F_{ij} = \frac{\partial}{\partial \mathbf{u}_j^n}\left(\frac{a_{ij}}{|V_i|}\mathbf{F}_{ij}(\mathbf{u}_i^n, \mathbf{u}_j^n)\cdot\mathbf{n}_{ij}\right), \tag{3.6}$$

and $\delta\mathbf{u_i}$ is the difference between the solution of the current and previous Newton iterations at cell i. The linear system (3.5) can be written as $Ax = b$, where

$$A_{ij} = \begin{cases} M_{ii} - \sum_{j\in\mathcal{N}(i)} F_{ji}, & j = i, \\ F_{ij}, & j \in \mathcal{N}(i), \\ 0, & \text{otherwise}, \end{cases} \tag{3.7}$$

$x = \delta\mathbf{u}$ is the difference between the solution of the current and previous Newton iterations, and $b = -\mathbf{\Psi}$, the residual of the Newton iteration.

3.1.1 Description of the linearized system

The coupled system with system matrix (3.7) can be partitioned in blocks in different manners. Let k be the number of non-pressure unknowns such that the system (3.1) consists of $k + 1$ equations. As mentioned in Chapter 2, there is a choice about which saturation and concentrations can be eliminated via constraints and phase equilibrium equations. Furthermore, in the construction of the Jacobian in (3.7), there is a choice about how to align the equations with unknowns. A mass conservation equation is typically aligned in the linearized system with the pressure unknowns. That equation is commonly referred to as the "pressure equation". Alternatively, the pressure equation can be a linear combination of the mass conservation equations.

We consider a linearized system of the following form: given a residual b, we search for x such that

$$Ax = \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} \begin{bmatrix} x_p \\ x_s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} = b, \tag{3.8}$$

where $A_{pp} \in \mathbb{R}^{N_p\times N_p}$ represents the pressure block coefficients, $A_{ss} \in \mathbb{R}^{N_s\times N_s}$ represents the secondary, non-pressure unknown blocks, and the rectangular blocks $A_{ps} \in \mathbb{R}^{N_p\times N_s}$ and $A_{sp} \in \mathbb{R}^{N_s\times N_p}$ represent the respective coupling coefficients. N_p is the number of discrete pressure variables, and $N_s = k\,N_p$ is the number of discrete non-pressure variables. Correspondingly, we denote the pressure solution and residual

as x_p and b_p, respectively, and the non-pressure solution and residual as x_s and b_s, respectively.

The ordering of the system (3.8) is done unknown-wise, i.e.

$$A = \begin{bmatrix} A_{pp}^{1,1} & \cdots & A_{pp}^{1,N_p} & A_{ps}^{1,1} & \cdots & A_{ps}^{1,N_p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{pp}^{N_p,1} & \cdots & A_{pp}^{N_p,N_p} & A_{ps}^{N_p,1} & \cdots & A_{ps}^{N_p,N_s} \\ A_{sp}^{1,1} & \cdots & A_{sp}^{1,N_p} & A_{ss}^{1,1} & \cdots & A_{ss}^{1,N_p} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ A_{sp}^{N_p,1} & \cdots & A_{sp}^{N_s,N_p} & A_{ss}^{N_p,1} & \cdots & A_{ss}^{N_p,N_p} \end{bmatrix}, \tag{3.9}$$

where $A_{pp}^{i,j} \in \mathbb{R}$, $A_{ss}^{i,j} \in \mathbb{R}^{k\times k}$, $A_{ps}^{i,j} \in \mathbb{R}^{1\times k}$ and $A_{sp}^{i,j} \in \mathbb{R}^{k\times 1}$ represent the coefficients of the system between grid cells i and j. Denoting each matrix entry as $a_{x,y}^{i,j}$, we can define the following for the grid cells i, j:

$$\begin{aligned} A_{pp}^{i,j} &= a_{1,1}^{i,j}, & A_{ps}^{i,j} &= \begin{bmatrix} a_{1,2}^{i,j} & \cdots & a_{1,k+1}^{i,j} \end{bmatrix}, \\ A_{sp}^{i,j} &= \begin{bmatrix} a_{2,1}^{i,j} \\ \vdots \\ a_{k+1,1}^{i,j} \end{bmatrix}, & A_{ss}^{i,j} &= \begin{bmatrix} a_{2,2}^{i,j} & \cdots & a_{2,k+1}^{i,j} \\ \vdots & \ddots & \vdots \\ a_{k+1,2}^{i,j} & \cdots & a_{k+1,k+1}^{i,j} \end{bmatrix}. \end{aligned} \tag{3.10}$$

Note that by using the (cell-centered) Finite Volume method, all unknowns are colocated. Therefore, as an alternative to (3.9), one could consider a point-wise ordering of the system as in (3.7), i.e.

$$A_{\text{point-wise}} = \begin{bmatrix} A^{1,1} & \cdots & A^{1,N_p} \\ \vdots & \ddots & \vdots \\ A^{N_p,1} & \cdots & A^{N_p,N_p} \end{bmatrix} = \begin{bmatrix} A_{pp}^{1,1} & A_{ps}^{1,1} & \cdots & A_{pp}^{1,N_p} & A_{ps}^{1,N_p} \\ A_{sp}^{1,1} & A_{ss}^{1,1} & & A_{sp}^{1,N_p} & A_{ss}^{1,N_p} \\ \vdots & & \ddots & & \vdots \\ A_{pp}^{N_p,1} & A_{ps}^{N_p,1} & & A_{pp}^{N_p,N_p} & A_{ps}^{N_p,N_p} \\ A_{sp}^{N_p,1} & A_{ss}^{N_p,1} & \cdots & A_{sp}^{N_p,N_p} & A_{ss}^{N_p,N_p} \end{bmatrix}, \tag{3.11}$$

and the corresponding reordering of x and b. These different orderings may be useful depending on the considered linear solver.

Notice from (3.7) that the blocks in (3.8) can be partitioned as

$$A = \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} M_{pp} & M_{ps} \\ M_{sp} & M_{ss} \end{bmatrix} + \begin{bmatrix} F_{pp} & F_{ps} \\ F_{sp} & F_{ss} \end{bmatrix} = \frac{1}{\Delta t} M + F, \tag{3.12}$$

where M and F represent the mass (or energy) accumulation and flux terms, respectively. The blocks of M are diagonal.

3.2 Krylov subspace methods

The most computationally expensive part of a reservoir simulator tends to be solving linearized systems of the form (3.8). It is therefore crucial to have robust and efficient linear solvers. The size of the linearized systems is often very large, but they are very sparse, which justifies the use of iterative methods over direct methods.

The most popular iterative linear solvers are Krylov subspace methods [86]. To approximate the solution of $Ax = b$, these methods construct a sequence of Krylov subspaces, $\mathcal{K}_n = \text{span}\left(\{b, Ab, A^2b, \ldots, A^{n-1}b\}\right)$. The prototypical Krylov subspace method is the conjugate gradient (CG) method, which is meant for symmetric positive-definite (spd) systems. The approximate solution x_n is constructed such that the residual $r_n = Ax_n - b$ is orthogonal to $\mathcal{K}_n$. However, we cannot usually assume spd properties for (3.8), and so we must consider methods for general linear systems.

The generalized minimal residual method (GMRES) [87] is a Krylov subspace method suitable for general linear systems. The approximate solution x_n is formed by minimizing the Euclidean norm of the residual r_n over the subspace $\mathcal{K}_n$. In GMRES, as opposed to CG, the subspaces are constructed explicitly. This entails that the amount of work and storage required per iteration increases linearly with the number of iterations. A common way to tackle this limitation is by restarting the algorithm after a number of iterations, say l, using the intermediate result as initial data for the next l iterations. This modification is called restarted GMRES and is denoted GMRES(l). A good choice of the parameter l can decrease solution time, while a bad choice can hinder or prevent convergence. Unfortunately, there is no obvious way to determine good values of l apart from experience and memory requirements. GMRES(l) is routinely used in reservoir simulation, where l is usually set to 20 or 30. When convergence fails, smaller time-steps can be taken to provide linearized systems which are easier to solve with GMRES.

An alternative to GMRES with less memory requirement is the biconjugate gradient method (Bi-CG). Instead of constructing an orthogonal sequence of residuals, Bi-CG constructs two mutually orthogonal sequences, which no longer provide minimization. Since this method can have an irregular convergence behaviour, a more robust alternative, the biconjugate gradient stabilized method (Bi-CGSTAB) [103], was developed. In this version, each Bi-CG step is followed by a GMRES(1) step. Similarly, for Bi-CGSTAB(l) [93], a GMRES(l) step follows every l Bi-CG steps. In this thesis, we will only consider GMRES.

3.3 Preconditioning

The rate of convergence of Krylov subspace methods depends on the properties of the linear system. For symmetric matrices, descriptive convergence bounds with respect to the spectrum of the system matrix are available. In the general case, however, it is harder to predict the convergence of GMRES. Convergence bounds based on eigenvalues with the eigenvector condition number, the field of values, and pseudospectra are often not descriptive [33]. Nevertheless, one would generally always wish to improve these system properties by preconditioning the linear system [108].

One way to build a preconditioner for the system $Ax = b$ is to find a matrix P which approximates A in some meaningful way, and for which the action of its inverse is easy to compute. Alternatively, approximate inverse preconditioners construct approximations of P^{-1} directly. The left-preconditioned system $P^{-1}Ax = P^{-1}b$ should be easier to solve using an iterative method. Similarly, we can right-precondition the system by solving $AP^{-1}y = b$, and then computing $x = P^{-1}y$. Right-preconditioning has the advantage of leaving the right-hand side intact, which allows for more meaningful convergence stopping criteria when based on the residual, which is the case for GMRES.

Note that a preconditioner does not have to be defined explicitly as a matrix; only a linear operation is required (i.e. the action of P^{-1}). For example, an iterative method could be used as a preconditioner for another iterative method.

For large scale computations such as reservoir simulation, preconditioners which are parallelizable or inherently parallel are very important. We will consider the scalability of different preconditioners.

Preconditioning strategies for systems of PDEs often involve the combination of different treatments for the different equations in the system [32, 108]. Indeed, the equations may have very distinct properties calling for different preconditioning strategies. In the case of reservoir simulation, the equations are elliptic with respect to the pressure, and hyperbolic with respect to the saturations[1], and the energy equation is parabolic with respect to the temperature (can be advection- or diffusion-dominated). Accordingly, a combination of strategies is needed to tackle these different properties.

In this section, we begin by a brief description of Incomplete LU factorization and Algebraic Multigrid methods in Sections 3.3.1 and 3.3.2, respectively. These

[1]The equations can be parabolic with respect to saturations when capillary pressure is not ignored, but still often advection-dominated.

methods are used as part of the industry-standard Constrained Pressure Residual method (CPR), detailed in Section 3.3.3. In Section 3.3.3.1, we describe decoupling operators standardly used in conjunction with CPR. We then present an alternative decoupling operator in Section 3.3.3.2.

3.3.1 Incomplete LU factorization (ILU)

The obvious way to solve the system $Ax = b$ directly is by first computing the LU decomposition $A = LU$ using Gaussian elimination, then simply solving the triangular systems $Ly = b$ and $Ux = y$ through forward and backward substitution. However, even if A is sparse, L and U may be much less sparse, which makes the factorization and triangular solves more expensive as well as significantly increasing the memory requirement. Instead, we can find sparse triangular matrices $\tilde{L}$ and $\tilde{U}$ such that $A \approx \tilde{L}\tilde{U}$. Solving $\tilde{L}\tilde{U}x = b$ is cheaper and provides an approximate solution to the original system. Using $\tilde{L}\tilde{U}$ as a preconditioner is known as Incomplete LU factorization (ILU) [66, 86].

A popular way to determine the sparsity pattern of $\tilde{L}$ and $\tilde{U}$ is to simply choose the relevant triangular parts of the sparsity pattern of A. This is known as ILU(0). More generally, choosing the sparsity pattern of A^{k+1} is called ILU(k). Other variants of ILU drop elements from the Gaussian elimination process using their magnitude rather than their location using a threshold — such approaches are denoted ILUT.

Both the processes of Gaussian elimination and triangular solves are inherently serial. Different domain decomposition strategies [31] can be used to parallelize ILU. Instead of being applied to the full problem, ILU can be applied to subdomains. The simple domain decomposition case without overlap corresponds to a block Jacobi preconditioner where ILU is used on the independent subproblems in parallel. Of course, the quality of the approximation decreases as the number of subdomains increases.

A recent development is the fine-grained parallel ILU [24], for which the coefficients of $\tilde{L}$ and $\tilde{U}$ are computed approximately, in parallel. This, combined with parallel approximate triangular solves, promises good parallel scalability for ILU. The efficiency of this method remains to be seen in the context of reservoir simulation.

Nested Factorization (NF) is a version of ILU developed specifically for reservoir simulation [6]. It differs from the more commonly used ILU factorizations in that it does not form the preconditioner from strictly upper and lower triangular factors. Instead, it constructs lower and upper factors using a procedure that adds one dimension at a time to the preconditioner. The convergence of GMRES when using NF is

sensitive to the ordering of the cells. Indeed, ordering the cells first along the direction of highest transmissibility (defined later in (3.55)) and then successively along directions of decreasing transmissibility usually provides the best results. Therefore, NF should nearly always be ordered first in the vertical direction.

3.3.2 Algebraic Multigrid (AMG)

Multigrid methods [14, 16, 102] use hierarchies of coarse grid approximations in order to solve differential equations. Smoothing/relaxation operations (such as Jacobi or Gauss-Seidel iterations) are combined with coarse grid corrections on increasingly coarser grids. For positive definite elliptic PDE operators, it is known that multigrid methods can provide optimal solvers (in the sense of linear scalability with the dimension of the discretized problem).

Algebraic Multigrid (AMG) [85, 98] uses information from the entries of the system matrix rather than that of the geometric grid. This makes AMG an ideal black-box solver for elliptic problems. Although it can be used to solve simpler problems, AMG is often used as a preconditioner for Krylov subspace methods in problems which are essentially elliptic but have additional characteristics.

For classical AMG, the construction of the coarse problems are based on heuristics usually assuming that the system matrix is an M-matrix (defined in Appendix A). M-matrices typically arise from the discretization of elliptic operators using low-order methods (Finite Volumes, Finite Elements, Finite Differences). High-order methods generally do not result in M-matrices, but AMG heuristics can often still be applied successfully with additional work. In the case of Finite Volume methods for reservoir simulation, the low order TPFA usually results in M-matrices for the pressure block. The higher-order multipoint flux approximation (MPFA) [1] results in smaller positive diagonal entries and positive off-diagonal, both of which can hinder the performance of AMG.

The key concept of the heuristics described in [85] is the determination of variables with "strong connections" by comparing the entries of the rows of the system matrix. Generally, we assume positive diagonal entries and nonpositive off-diagonal entries (as in an M-Matrix). If the coefficients of the system matrix A are a_{ij}, then the index i is considered "strongly connected" to index j if

$$-a_{ij} \geq \theta \max_{k \neq i} |a_{ik}|, \tag{3.13}$$

where θ is some "strength parameter" typically between 0.2 and 0.25. Then, for each index i, we have the set of indices for which i is strongly connected, $S_i =$

$\{j : i \text{ is strongly connected to } j\}$, and the set of indices strongly connected to i, $S_i^\top = \{j : i \in S_j\}$. The number of entries in $S_i^\top$, $|S_i^\top|$, determines the importance of variable i, and whether or not it should be included in the coarse space. By cycling through the indices, the variables are then separated into a coarse space and a fine space.

In the case of symmetric positive-definite systems, theoretical convergence results of AMG are well understood, especially for M-matrices [96]. In the case of non-symmetric M-matrices, some theoretical convergence results also hold [70]. Classical AMG can (and often does) fail if the system matrix is not an M-matrix. In practice, convergence of AMG is observed in cases where violations of M-matrix properties are relatively small.

Relative to preconditioners such as ILU, parallel variants of multigrid methods (such as BoomerAMG [46]) retain more effectiveness. The commercial reservoir simulator Intersect [30] has a parallel implementation of AMG [72].

Classical AMG approaches are known to provide efficient solvers for scalar PDEs. However, these are based on a variable-based approach not distinguishing between physical unknowns. Extensions of classical AMG to efficiently solve systems of PDEs use unknown-based or point-based approaches. See [26] for a detailed description and a general framework for such methods.

The unknown-based strategy [84] is to create coarse grids for the variables corresponding to the same unknown separately. For a matrix with an unknown-wise ordering, classical AMG coarsening and interpolation is used on the diagonal blocks to determine the coarse grid. Note that this differs from applying AMG to the diagonal blocks independently since the coarse problem includes the off-diagonal blocks. Computationally cheap and easy to implement, the unknown-based approach will perform well if the cross-coupling between unknowns is not too strong and the diagonal blocks are amenable for the application of classical AMG.

With a typical Finite Volume scheme, degrees of freedoms associated with the different variables are located at the same points. This means that one could consider using a nodal/point-based AMG method (initially designed for linear elasticity problems [84]). Instead of considering all degrees of freedoms individually, they are grouped in a vector at each "point". The construction of the coarse problems uses similar heuristics to the scalar equivalent but applied to the "point" vectors and resulting blocks in the system matrix. However, the performance of coarsening strategies for the point-based approach is problem-specific. Several strategies are discussed in [26].

Naively applying a point-based or unknown-based AMG method to the full reservoir simulation system does not work in the common case where the equations are hyperbolic with respect to the saturations. Substantial work is needed to construct meaningful coarse problems when applying AMG to the full system [27, 44]. The proposed methods include unknown-based AMG approaches, and decoupling operators focused on maintaining AMG-friendly properties.

However, the equations are elliptic with respect to pressure, and the energy equation which is parabolic with respect to temperature. We will therefore consider applying AMG to a pressure-temperature system for single-phase flow in Chapter 5, and a pressure-temperature subsystem from the multiphase case in Chapter 6.

Efficient implementations of AMG methods for systems include BoomerAMG [46] and multigrid reduction (MGR) from the hypre library [36], and Smoothed Aggregation from the ML package [39]. For example, BoomerAMG was effective in some diffusion-dominated two-phase flow problems (where capillary pressure was significant) [17], and MGR had some success with multiphase flow problems [18, 107].

3.3.3 Constrained Pressure Residual (CPR)

Isothermal models result in a linearized system of a mixed character. It consists of an elliptic-like problem for the pressure, and a hyperbolic-like problem for the saturations/concentrations. While these have a mostly local effect, the pressure has a global influence over the reservoir. The ellipticity of the pressure block requires good global preconditioning (such as AMG or NF) as opposed to the hyperbolic-like blocks where local preconditioning is sufficient (e.g. ILU(0)).

The Constrained Pressure Residual (CPR) method [105, 106] is the industry-standard preconditioner for reservoir simulation. This method was initially developed for isothermal multiphase models such as the one described in Section 2.2.1, but is also applied to compositional and thermal cases. In the thermal case, temperature is in practice grouped with the saturation/concentration unknowns as secondary unknowns. Pressure is considered the primary unknown.

The system (3.8) is preconditioned in a two-stage process. First, a restricted pressure system is solved, followed by the application of a preconditioner to the full system. Given the residual $b = \begin{bmatrix} b_p & b_s \end{bmatrix}^\top$, the CPR two-stage preconditioner provides a correction as follows:

1. Approximately solve the pressure system (e.g. using one AMG V-cycle):

$$A_{pp}\delta_p^0 = b_p; \tag{3.14}$$

2. Compute the new residual:

$$\begin{bmatrix} r_p \\ r_s \end{bmatrix} = \begin{bmatrix} b_p \\ b_s \end{bmatrix} - \begin{bmatrix} A_{pp} \\ A_{sp} \end{bmatrix} \delta_p^0; \tag{3.15}$$

3. Precondition and correct (e.g. using ILU(0)):

$$\begin{bmatrix} \delta_p \\ \delta_s \end{bmatrix} = P^{-1} \begin{bmatrix} r_p \\ r_s \end{bmatrix} + \begin{bmatrix} \delta_p^0 \\ 0 \end{bmatrix}, \tag{3.16}$$

where $\delta = \begin{bmatrix} \delta_p & \delta_s \end{bmatrix}^\top$ is the correction obtained after the two-stage process. Then, the effect of the two-stage preconditioner can be written as

$$\delta = P^{-1} \left(I - (A - P) \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) b. \tag{3.17}$$

The elliptic nature of the pressure block makes it a natural candidate for multigrid methods. The version of CPR using AMG in the first stage, often denoted CPR-AMG, was proposed in [54, 55]. It is now widely used in reservoir simulation as a right preconditioner for GMRES, where a single AMG V-cycle for the pressure system is standard.

CPR is a multi-stage preconditioner. More generally, given preconditioners $P_1, \ldots, P_k$, a k-stage preconditioner can be written as

$$\begin{aligned} P^{-1} = {} & P_k^{-1}(I - AP_{k-1}^{-1}) \ldots (I - AP_1^{-1}) + \ldots \\ & + P_3^{-1}(I - AP_2^{-1})(I - AP_1^{-1}) + P_2^{-1}(I - AP_1^{-1}) + P_1^{-1}. \end{aligned} \tag{3.18}$$

In particular, a two-stage preconditioner can be written as

$$P^{-1} = P_2^{-1}(I - AP_1^{-1}) + P_1^{-1}. \tag{3.19}$$

For the CPR preconditioner, P_1^{-1} is the restricted pressure approximate solver

$$P_1^{-1} = \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \tag{3.20}$$

where the inverse of the pressure block is approximated using an AMG V-cycle in CPR-AMG, and P_2^{-1} is the full system approximate solver. Of course, P_1^{-1} is not an invertible matrix.

Let R_p be the pressure restriction operator such that $R_p A R_p^\top = A_{pp}$. It can be useful (for example when considering an unknown-wise ordering) to rewrite (3.20) as

$$P_1^{-1} = R_p^\top (R_p A R_p^\top)^{-1} R_p. \tag{3.21}$$

This is analogous to the use of restriction and extension operators in domain decomposition, although here it restricts to variables spaces rather than subdomains.

The CPR preconditioner was originally designed for isothermal models. In the thermal case, temperature is often considered as a secondary unknown along with the other non-pressure unknowns and preconditioned in the second stage of CPR. The temperature is not as local as saturations and concentrations and so, in practice, ILU(1) is usually required instead of ILU(0) in (3.16). This is a consequence of heat diffusion being present in the energy conservation equation. In Chapter 6, we will present an extension of CPR where heat diffusion is treated appropriately.

3.3.3.1 Decoupling operators

The CPR preconditioner is usually used in conjunction with so-called decoupling operators. These are used to reduce the coupling between the pressure equation and and secondary unknowns, i.e. A_{ps}, (and sometimes between the secondary equations and pressure, i.e. A_{sp}). For a generic approximate decoupling operator G, we obtain an approximately decoupled system

$$GAx = \tilde{A}x = \begin{bmatrix} \tilde{A}_{pp} & \tilde{A}_{ps} \\ \tilde{A}_{sp} & \tilde{A}_{ss} \end{bmatrix} \begin{bmatrix} x_p \\ x_s \end{bmatrix} = \begin{bmatrix} \tilde{b}_p \\ \tilde{b}_s \end{bmatrix} = \tilde{b} = Gb. \tag{3.22}$$

Decoupling operators are often applied to the linear system as a preprocessing step. Alternatively, decoupling operators G are sometimes used within the first stage of CPR as follows:

$$P_1^{-1} = R_p^\top (R_p G A R_p^\top)^{-1} R_p G. \tag{3.23}$$

If G is used as a left preconditioner, it also changes the residual b. To prevent this effect, it is desirable that $\|G\|$ be close to one. In practice, this can be achieved by normalizing the rows of G.

Some desirable properties for $\tilde{A}$ are listed in [99]. Especially for CPR-AMG, it is important that $\tilde{A}_{pp}$ remains close to an M-matrix.

The actual pressure equation in (3.8) is $A_{pp}x_p + A_{ps}x_s = b_p$, which differs from the one approximately solved in the first-stage of CPR (3.14). An approximation is made in this stage because the coupling with the secondary unknowns is ignored. It would then be desirable that A_{ps} be "small" for this approximation to be good.

We can decouple the pressure equation exactly by performing a Schur complement decomposition

$$\begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} = \begin{bmatrix} I & A_{ps}A_{ss}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp} - A_{ps}A_{ss}^{-1}A_{sp} & 0 \\ A_{sp} & A_{ss} \end{bmatrix}, \tag{3.24}$$

such that by inverting the upper triangular matrix

$$\begin{bmatrix} I & -A_{ps}A_{ss}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} = \begin{bmatrix} A_{pp} - A_{ps}A_{ss}^{-1}A_{sp} & 0 \\ A_{sp} & A_{ss} \end{bmatrix}, \tag{3.25}$$

we obtain the exact decoupling operator

$$G_{\text{exact}} = \begin{bmatrix} I & -A_{ps}A_{ss}^{-1} \\ 0 & I \end{bmatrix}. \tag{3.26}$$

Applying G_{exact} to the system (3.8) removes the coupling between the pressure equation and the non-pressure blocks. However, calculating A_{ss}^{-1} is not easy in general. Additionally, since A_{ss}^{-1} is generally dense, the Schur complement $S_p = A_{pp} - A_{ps}A_{ss}^{-1}A_{sp}$ will also be. The following decoupling operators construct $\tilde{A}_{pp}$ with a sparsity pattern similar to that of A_{pp}.

Alternate-Block Factorization (ABF) This first decoupling approach seeks to reduce the coupling in both the pressure equation and the secondary equations. The inverse of the ABF decoupling operator [10] (also known as block-diagonal inverse) is defined as

$$G_{\text{ABF}}^{-1} = \begin{bmatrix} D_{pp} & D_{ps} \\ D_{sp} & D_{ss} \end{bmatrix} = \begin{bmatrix} \text{diag}(A_{pp}) & \text{diag}(A_{ps}) \\ \text{diag}(A_{sp}) & \text{diag}(A_{ss}) \end{bmatrix}, \tag{3.27}$$

where diag() returns the block-diagonal of a block matrix. Applying G to the unknown-wise ordered A defined in (3.9) gives us

$$G_{\text{ABF}}A = \begin{bmatrix} \Delta^{-1} & 0 \\ 0 & \Delta^{-1} \end{bmatrix} \begin{bmatrix} D_{ss}A_{pp} - D_{ps}A_{sp} & D_{ss}A_{ps} - D_{ps}A_{ss} \\ D_{pp}A_{sp} - D_{sp}A_{pp} & D_{pp}A_{ss} - D_{sp}A_{ps} \end{bmatrix}, \tag{3.28}$$

where $\Delta = D_{pp}D_{ss} - D_{ps}D_{sp}$. It can easily be verified that the main diagonal entries of $\tilde{A}_{pp}$ and $\tilde{A}_{ss}$ are equal to one, and those of $\tilde{A}_{ps}$ and $\tilde{A}_{sp}$ are equal to zero. The pressure equation to be solved in the two-stage preconditioning is thus

$$\tilde{A}_{pp}x_p = \Delta^{-1}(D_{ss}A_{ps} - D_{ps}A_{ss})x_p = \tilde{b}_p. \tag{3.29}$$

We observe that, with respect to the point-wise ordering, we have the following decoupling operator:

$$(G_{\text{ABF}}^{\text{point-wise}})^{-1} = \begin{bmatrix} A^{1,1} & & \\ & \ddots & \\ & & A^{N_p,N_p} \end{bmatrix}. \tag{3.30}$$

We see that applying $G_{\text{ABF}}^{\text{point-wise}}$ to the system is equivalent to doing a block-diagonal scaling or a block Jacobi iteration. This decoupling operator modifies the coefficients of both the pressure and saturation equations. This will not be the case for the remaining decoupling operators in this chapter, which only modify coefficients in the pressure equation, i.e. the A_{pp} and A_{ps} blocks.

Quasi-IMPES (QI) The quasi-IMPES (QI) decoupling operator [54, 55] is defined as

$$G_{\text{QI}} = \begin{bmatrix} I & -\text{diag}(A_{ps})\text{diag}(A_{ss})^{-1} \\ 0 & I \end{bmatrix}. \tag{3.31}$$

Applying this to the system matrix of (3.9) gives us

$$G_{\text{QI}}A = \begin{bmatrix} A_{pp} - D_{ps}D_{ss}^{-1}A_{sp} & A_{ps} - D_{ps}D_{ss}^{-1}A_{ss} \\ A_{sp} & A_{ss} \end{bmatrix}. \tag{3.32}$$

This leads to the following pressure equation for CPR (3.14):

$$\tilde{A}_{pp}x_p = (A_{pp} - D_{ps}D_{ss}^{-1}A_{sp})x_p = \tilde{b}_p. \tag{3.33}$$

This decoupling has the clear advantage of leaving the coefficients of the secondary equations untouched. The main block-diagonal of $\tilde{A}_{ps}$ is also zeroed out. Indeed, the diagonal terms of $\tilde{A}_{ps}$ are given by $A_{ps}^{i,i} - A_{ps}^{i,i}(A_{ss}^{i,i})^{-1}A_{ss}^{i,i} = 0$.

Since the A_{ps} and A_{ss} blocks consist of partial derivatives from mostly local quantities (saturations/concentrations), they are highly diagonally dominant, which means that D_{ps} and D_{ss} can be meaningful approximations. Diagonal dominance is defined in Appendix A.

True-IMPES (TI) The previously presented decoupling operators are justified algebraically. The true-IMPES decoupling operator [54, 55] is an approach which can be justified by the structure of the conservation equations. Conveniently, it can still be defined algebraically.

Since the flux terms have very little dependence on secondary unknowns, the main contribution to the blocks A_{ps} and A_{ss} come from the mass accumulation part. Knowing this, a natural approximation would be

$$A_{ps}A_{ss}^{-1} \approx M_{ps}M_{ss}^{-1}, \tag{3.34}$$

where M_{ps} and M_{ss} are the mass accumulation blocks from (3.12), which are (block-) diagonal matrices. However, it may not be practical to explicitly define the splitting

of mass accumulation and flux terms. Thankfully, looking back at (3.7), we see that the mass accumulation blocks can be recovered by summing the columns, cancelling the flux terms. Thus, the TI decoupling operator is defined as

$$G_{\text{TI}} = \begin{bmatrix} I & -\text{colsum}(A_{ps})\text{colsum}(A_{ss})^{-1} \\ 0 & I \end{bmatrix}, \tag{3.35}$$

where colsum() returns a block-diagonal matrix, where each diagonal block is the sum of the blocks in the corresponding block column of the original matrix. Applying this to the system matrix of (3.8) gives us

$$G_{\text{TI}}A = \begin{bmatrix} A_{pp} - \text{colsum}(A_{ps})\text{colsum}(A_{ss})^{-1}A_{sp} & A_{ps} - \text{colsum}(A_{ps})\text{colsum}(A_{ss})^{-1}A_{ss} \\ A_{sp} & A_{ss} \end{bmatrix}. \tag{3.36}$$

This leads to the following pressure equation:

$$\tilde{A}_{pp}x_p = (A_{pp} - \text{colsum}(A_{ps})\text{colsum}(A_{ss})^{-1}A_{sp})x_p = \tilde{b}_p. \tag{3.37}$$

Full Row Sum (FRS) The following approach was inspired by IMPES in [90], where a pressure equation is constructed by summing the mass conservation equations together. It was first called quasi-IMPES, but when revisited in [42], this method was renamed Full Row Sum (FRS) to avoid confusion. The FRS decoupling operator, applied to the point-wise ordering of the system, is in the form

$$G_{\text{FRS}}^{\text{point-wise}} = \begin{bmatrix} G^1 & & \\ & \ddots & \\ & & G^{N_p} \end{bmatrix}, \tag{3.38}$$

where each block G^i is an upper triangular $(k+1) \times (k+1)$ matrix and defined as

$$G^i = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix}. \tag{3.39}$$

This method mostly focuses on constructing a descriptive pressure equation. In general, FRS does not provide a good decoupling. However, under some assumptions (e.g. the densities of the different phases are similar and the flow is close to incompressible), FRS should also provide a quality decoupling [90]. The obtained $\tilde{A}_{pp}$ block

is essentially the same as for IMPES, except for the time-step at which the secondary unknowns are evaluated. The pressure equation for FRS is given by

$$\tilde{A}_{pp}x_p = \left(A_{pp} + \sum_{i=1}^{k} A_{s_ip} \right) x_p = \tilde{b}_p, \tag{3.40}$$

where the A_{s_ip} blocks represent the pressure coupling for the i^{th} secondary equation.

Dynamic Row Sum (DRS) Similarly to FRS, the Dynamic Row Sum (DRS) preconditioner [42, 44] seeks a descriptive pressure equation, but does not automatically sum all equations. Instead, it dynamically chooses which equations to include in the pressure equation. DRS aims to improve the solvability of $\tilde{A}_{pp}x_p = \tilde{b}_p$ by AMG. The DRS decoupling operator, applied to the point-wise ordering of the system, is in the form (3.38) where each block G^i is an upper triangular matrix defined as

$$G^i = \begin{bmatrix} \delta_1^i & \delta_2^i & \delta_3^i & \cdots & \delta_{k+1}^i \\ & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \vdots \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix}, \tag{3.41}$$

where

$$\delta_x^i = \begin{cases} 0 & \text{if } \dfrac{a_{x,1}^{i,i}}{\sum_{j=1,j\neq i}^{N_p} |a_{x,1}^{i,j}|} < \epsilon_{dd}, \\ 1 & \text{otherwise,} \end{cases} \tag{3.42}$$

and $0 \leq \epsilon_{dd} \leq 1$.

The idea of this method is to only include in $\tilde{A}_{pp}$ the parts of A which do not introduce unwanted matrix properties with respect to the application of AMG. It constructs an IMPES-like pressure equation like FRS, but only sets $\delta_x^i = 1$ for equations with desirable properties. Indeed, the parameter ϵ_{dd} controls how much violation of diagonal dominance in $\tilde{A}_{pp}$ is allowed. For $\epsilon_{dd} = 1$, the DRS method attempts to build a weakly diagonally dominant $\tilde{A}_{pp}$. This, however, may slow down the convergence of the CPR-preconditioned GMRES. The authors of [44] suggest $\epsilon_{dd} = 0.2$ as a good compromise between the good matrix properties for AMG and the pressure equation being close to the one in IMPES schemes. They also include some safeguard checks. To prevent the ϵ_{dd} from being too small, the diagonal dominance of $\tilde{A}_{pp}$ is checked. If it is found to be unsatisfied, the δ_x^i are recomputed using a larger ϵ_{dd}. Also, if $\delta_1^i = 0$, G^i (and thus G) is singular. This is corrected in the following manner: if $\delta_x^i \neq 0$ for some $x > 1$, in the x^{th} row of G^i, the diagonal is moved to the first

column, and if $\delta_x^i = 0$ for all x, set $\delta_x^i = 1$, where x is the row that least violates diagonal dominance.

The DRS approach illustrates how there are competing goals for the decoupling operators used with CPR-AMG. There is a balance for the pressure equation between having a quality decoupling and desirable properties for AMG convergence. Decoupling operators such as QI and TI seek mainly to make $\tilde{A}_{ps}$ smaller. In practice, CPR-AMG performs well with these approaches for simple multiphase problems but sometimes fails to perform in more advanced compositional cases. In Section 3.4, we will discuss why this may be the case.

3.3.3.2 Least-squares decoupling operator

Reducing the norm of A_{ps} is what decoupling operators such as quasi-IMPES (QI) and true-IMPES (TI) seek to do. While these are usually viewed as preprocessing steps in the reservoir simulation community, they can be understood in a more general framework.

Instead of using the exact Schur complement, $A_{ps}A_{ss}^{-1}$ is approximated with (block-) diagonal matrices. We can generalize the previous decoupling approaches in the following manner:

$$GAx = \begin{bmatrix} I & -M \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{bmatrix} x = \begin{bmatrix} A_{pp} - MA_{sp} & A_{ps} - MA_{ss} \\ A_{sp} & A_{ss} \end{bmatrix} x = \begin{bmatrix} b_p - Mb_s \\ b_s \end{bmatrix}, \tag{3.43}$$

where M approximates $A_{ps}A_{ss}^{-1}$.

The QI decoupling operator uses $M = \text{diag}(A_{ps})\text{diag}(A_{ss})^{-1}$ while the TI decoupling operator uses $M = \text{colsum}(A_{ps})\text{colsum}(A_{ss})^{-1} = M_{ps}M_{ss}^{-1}$. As mentioned in [90] for the two-phase case, FRS provides a good decoupling if $A_{pp} + A_{sp}$ is a good approximation of the Schur complement, i.e. $M = -I$.

Least-squares decoupling operator (LSQ) As an alternative to the above, one can construct a decoupling operator that focuses on making the norm of $\tilde{A}_{ps}$ as small as possible. An obvious choice to make this block small is by minimizing its norm over the choice of M in (3.43). We investigated this approach in [81].

We consider the use of sparse approximate inverses (with target matrix) [47] in choosing the matrix M. The construction of M is given by

$$\min_{M \in S_M} \|A_{ps} - MA_{ss}\|, \tag{3.44}$$

where S_M denotes the set of matrices with a given sparsity pattern, i.e. for a set of indices $I_M \subseteq \{(i,j) \mid 1 \leq i \leq N_p, 1 \leq j \leq N_s\}$, we have that $S_M = \{M \in \mathbb{R}^{N_p \times N_s} \mid M_{i,j} = 0 \text{ for } (i,j) \notin I_M\}$, and $\|.\|$ denotes the Frobenius norm. The row-by-row nature of the Frobenius norm enables us to write

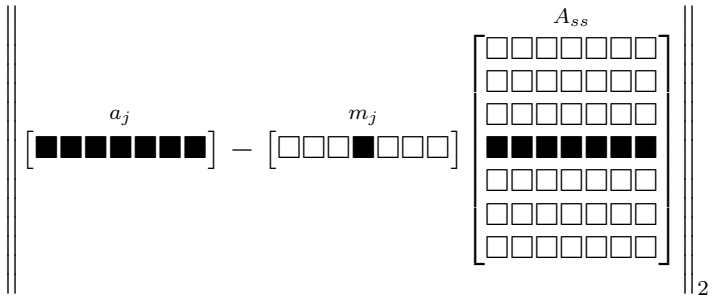
$$\min_{M \in S_M} \|A_{ps} - MA_{ss}\|^2 = \sum_{j=1}^{N_p} \min_{m_j \in S_{m_j}} \|a_j - m_j A_{ss}\|_2^2, \tag{3.45}$$

where a_j and m_j denote the j^{th} rows of A_{ps} and M, respectively, S_{m_j} is the set of j^{th} rows of matrices in S_M, and $\|.\|_2$ is the Euclidean norm. As all rows are independent, this procedure is easily parallelizable.

We now consider different choices for the sparsity pattern S_M. Obviously, taking S_M as the set of all matrices in $\mathbb{R}^{N_p \times N_s}$ gives us $M = A_{ps}A_{ss}^{-1}$, which results in an exact pressure decoupling, i.e. $\tilde{A}_{ps} = 0$. This, however, causes much fill (additional non-zero entries) in $\tilde{A}_{pp} = A_{pp} - MA_{sp}$. Therefore, we seek M that preserves the sparsity pattern of A_{pp}.

A (block-) diagonal matrix preserves the sparsity pattern of any (block) matrix it is multiplied to. Thus, we choose $I_M = \{(i,j) \mid 1 \leq i \leq N_p, (i-1)k+1 \leq j \leq ik\}$ and get that S_M is the set of block-diagonal matrices with blocks of size $1 \times k$. Obviously, any larger set could cause additional fill in the resulting $\tilde{A}_{pp}$ and $\tilde{A}_{ps}$.

Looking at the j^{th} least squares problem as formulated in (3.45), we are minimizing over the values of the block $M_{j,j}$, i.e. over k variables, and taking the other values in m_j as null. This can be illustrated as follows, where black squares denote possibly non-empty blocks and white squares are empty or unused blocks:



. (3.46)

Of course, the product of the sparse row vector m_j with A_{ss} is equivalent to the product of $M_{j,j}$ with the j^{th} row of A_{ss}, denoted $A_{ss}^{j,*}$. The j^{th} minimization problem becomes

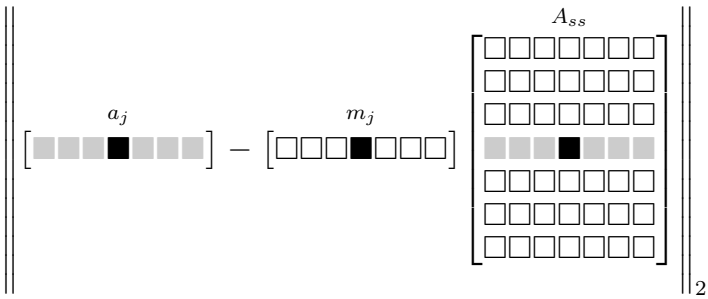
$$\min_{M_{j,j} \in \mathbb{R}^k} \|a_j - M_{j,j}A_{ss}^{j,*}\|_2. \tag{3.47}$$

The well-known least squares solution is

$$M_{j,j} = (a_j A_{ss}^{j,*\top})(A_{ss}^{j,*} A_{ss}^{j,*\top})^{-1}. \tag{3.48}$$

We call this particular choice of M the LSQ decoupling operator. Note that the block sparsity patterns of A_{sp} and A_{pp} should be similar and thus adding linear combinations of the rows of A_{sp} to A_{pp} should add very little fill.

We can consider a different minimization problem where A_{ps} and A_{ss} in (3.44) are replaced with restrictions of these matrices on an equivalent sparsity pattern to S_M, i.e. for the diagonal of A_{ps} and square block-diagonal of A_{ss}. Denoting ignored blocks as grey squares, we illustrate this as follows:



(3.49)

We now only need the main diagonal blocks as opposed to whole rows of the matrices. The j^{th} minimization problem becomes

$$\|A_{ps}^{j,j} - M_{j,j}A_{ss}^{j,j}\|_2. \tag{3.50}$$

The solution to this problem is

$$M_{j,j} = (A_{ps}^{j,j}A_{ss}^{j,j\top})(A_{ss}^{j,j}A_{ss}^{j,j\top})^{-1} = A_{ps}^{j,j}(A_{ss}^{j,j})^{-1}, \tag{3.51}$$

which is the same as the QI decoupling operator.

Comparison with QI and TI The LSQ decoupling operator is similar to QI and TI in that their main objective is to reduce the norm of $\tilde{A}_{ps}$. Their behaviour will be similar in many cases. In fact, looking at (3.12), as Δt goes to zero, it is easy to see that the three methods become equivalent. Indeed, in this asymptotic case, the mass accumulation term dominates so that A_{ps} and A_{ss} become $\frac{1}{\Delta t}M_{ps}$ and $\frac{1}{\Delta t}M_{ss}$, respectively.

In terms of computational cost, all three methods are rather similar (the decoupling step is cheap relative to the whole CPR preconditioning process). LSQ offers very few advantages over the classical alternatives. The only observed cases where it performs slightly better are ones where QI and TI already perform well (see results in the technical report [81]). However, in cases that are more challenging for QI and TI, LSQ fails to perform. By focusing on reducing the norm of $\tilde{A}_{ps}$, it appears that unwanted properties have been introduced in the system. We will discuss why this may be the case in Section 3.4.

Balanced decoupling strategy We later discovered that the LSQ decoupling operator was similar to the approach proposed in [3]. The balanced decoupling strategy (BDS) is a decoupling operator of the form $G = \overline{\overline{C}}\,\overline{C}$, where

$$\overline{C} = \begin{bmatrix} I & 0 \\ C_1 & C_2 \end{bmatrix}, \qquad \overline{\overline{C}} = \begin{bmatrix} C_3 & C_4 \\ 0 & I \end{bmatrix}, \tag{3.52}$$

$$\overline{A} = \overline{C}A = \begin{bmatrix} A_{pp} & A_{ps} \\ C_1A_{pp} + C_2A_{sp} & C_1A_{ps} + C_2A_{ss} \end{bmatrix} = \begin{bmatrix} A_{pp} & A_{ps} \\ \overline{A}_{sp} & \overline{A}_{ss} \end{bmatrix}, \tag{3.53}$$

$$\overline{\overline{A}} = \overline{\overline{C}}\,\overline{A} = \begin{bmatrix} C_3A_{pp} + C_4\overline{A}_{sp} & C_3A_{ps} + C_4\overline{A}_{ss} \\ \overline{A}_{sp} & \overline{A}_{ss} \end{bmatrix} = \begin{bmatrix} \overline{\overline{A}}_{pp} & \overline{\overline{A}}_{ps} \\ \overline{A}_{sp} & \overline{A}_{ss} \end{bmatrix}. \tag{3.54}$$

The block-diagonal matrices C_1 and C_2 are computed using least squares to minimize $\|\overline{A}_{sp}\|/\|\overline{A}_{ss}\|$. Similarly, C_3 and C_4 minimize $\|\overline{\overline{A}}_{ps}\|/\|\overline{\overline{A}}_{pp}\|$. In particular, the authors use $\overline{\overline{C}}$ on its own, with $C_3 = I$, which is equivalent to the LSQ decoupling operator.

3.4 Properties of the block matrices

We now investigate the properties of the different blocks of the system (3.8). We are interested in diagonal dominance and M-matrix properties. A high diagonal dominance can justify the use of ILU or similar preconditioners. The M-matrix properties are important for the applicability of AMG to the pressure block A_{pp} (or $\tilde{A}_{pp}$ when using decoupling operators). The properties of other blocks are important since they are used to construct $\tilde{A}_{pp}$.

In his thesis [42], Gries performs a study of the properties of the different blocks of (3.8). His analysis focuses on a 1D black-oil model with Finite Differences for spatial discretization and TPFA for the flux terms. The conclusions are then extended to more advanced models. We will discuss in Section 3.4.3 that some of these properties may not extend to the compositional case, and in Section 3.4.4, the resulting consequences with the use of traditional decoupling operators.

3.4.1 Multiphase flow

As done in [42], we first consider isothermal multiphase flow as described by (2.18) in 1D with spatial discretization via Finite Differences. Each cell i has neighbouring cells $i-1$ and $i+1$ with identical surface A between each cell, and distance h between the centers of each cell. Recall that K denotes the permeability, ρ_α, the density of phase α, μ_α, its viscosity, and $k_{r\alpha}$, its relative permeability. The transmissibility of phase α between two cells is defined as

$$[T_\alpha]^n_{i\pm\frac{1}{2}} = \frac{K_{i,i\pm\frac{1}{2}}A}{h}\frac{[\rho_\alpha]^n_{i\pm\frac{1}{2}}}{[\mu_\alpha]^n_{i\pm\frac{1}{2}}}[k_{r\alpha}]^n_{i\pm\frac{1}{2}}, \tag{3.55}$$

where $[\ ]^n_i$ denotes the discretized quantities at time-step n for cell i. Quantities at $[\ ]_{i\pm\frac{1}{2}}$ can be calculated as the average quantity between cells i and $i+1$ or with upstream weighting, i.e. using the pressure p_α of the previous time-step (or Newton iteration) as follows:

$$i_\alpha^{\text{upstream}} = \begin{cases} i \pm 1 & \text{if } [p_\alpha]^{n-1}_{i\pm1} > [p_\alpha]^n_{i\pm1}, \\ i & \text{if } [p_\alpha]^n_{i\pm1} \geq [p_\alpha]^{n-1}_{i\pm1}. \end{cases} \tag{3.56}$$

Recall that ϕ is the porosity of the porous medium, S_α is the saturation of phase α, and q_α is the injection/production rate so that $\rho_\alpha q_\alpha$ is the mass source/sink term. Using backward Euler for temporal discretization, the discretized version of (2.18) for phase α in cell i and time-step n is

$$\begin{aligned}[A_\alpha]^n_i :=& \frac{1}{\Delta t}\left([\phi\rho_\alpha S_\alpha]^n_i - [\phi\rho_\alpha S_\alpha]^{n-1}_i\right) \\ &- \frac{1}{hA}\left([T_\alpha]^n_{i-\frac{1}{2}}([p_\alpha]^n_{i-1} - [p_\alpha]^n_i) + [T_\alpha]^n_{i+\frac{1}{2}}([p_\alpha]^n_{i+1} - [p_\alpha]^n_i)\right) \\ &- [\rho_\alpha q_\alpha]^n_i,\end{aligned} \tag{3.57}$$

where we have ignored gravity for simplicity. For a model with phases α, β, and γ, we can construct the system (3.8) by taking partial derivatives of (3.57) with respect to a chosen pressure $[p_\alpha]^n_i$ and saturations $[S_\alpha]^n_i$, $[S_\beta]^n_i$, and $[S_\gamma]^n_i$. Aligning the α equation to be the pressure equation, and eliminating S_α with the saturation constraint (2.14), we get

$$\begin{aligned} A_{pp} &= \left[\frac{\partial[A_\alpha]^n_i}{\partial[p_\alpha]^n_i}\right], & A_{ps} &= \begin{bmatrix} \frac{\partial[A_\alpha]^n_i}{\partial[S_\beta]^n_i} & \frac{\partial[A_\alpha]^n_i}{\partial[S_\gamma]^n_i} \end{bmatrix}, \\ A_{sp} &= \begin{bmatrix} \frac{\partial[A_\beta]^n_i}{\partial[p_\alpha]^n_i} \\ \frac{\partial[A_\gamma]^n_i}{\partial[p_\alpha]^n_i} \end{bmatrix}, & A_{ss} &= \begin{bmatrix} \frac{\partial[A_\beta]^n_i}{\partial[S_\beta]^n_i} & \frac{\partial[A_\beta]^n_i}{\partial[S_\gamma]^n_i} \\ \frac{\partial[A_\gamma]^n_i}{\partial[S_\beta]^n_i} & \frac{\partial[A_\gamma]^n_i}{\partial[S_\gamma]^n_i} \end{bmatrix}. \end{aligned} \tag{3.58}$$

Under reasonable assumptions such as Δt being small enough, and disregarding wells for now, Gries states some properties of these blocks. He concludes that A_{pp} and the blocks of A_{sp} are M-matrices which are almost symmetric. Also, the diagonal blocks of A_{ss} as well as the blocks of $-A_{ps}$ are strongly diagonally dominant M-matrices. This negative sign results from setting $S_\alpha = 1 - S_\beta - S_\gamma$.

In this case, the off-diagonal blocks of A_{ss} have no contribution from the mass accumulation term since ρ_β only depends on p_β and ϕ is usually constant with respect to S_γ. The off-diagonal blocks depend on the flux term in how the relative permeability varies with respect to the other saturations. According to [42], the off-diagonal blocks are expected to be Z-matrices (defined in Appendix A), with no assumption on their diagonal dominance. In observed cases, however, the off-diagonal blocks are simply zero or diagonal matrices with negative entries.

Influence of injection wells Wells only contribute to the diagonal terms of the pressure-related blocks of (3.58). The derivative of density with respect to pressure is positive. Therefore, production wells ($q_\alpha < 0$) result in a positive contribution to the diagonal.

In contrast, injection wells ($q_\alpha > 0$) result in a negative contribution to the diagonal. Consequently, several rows of A_{pp} (and the blocks of A_{sp}) may violate diagonal dominance. By losing M-matrix properties and acquiring negative eigenvalues, A_{pp} may no longer be suitable for AMG.

One way to circumvent the effects of injection wells is by taking very small timesteps during the injection phase of a simulation. An increase in the size of the mass accumulation term is used to counter the negative contribution of injection wells. Additionally, the DRS decoupling operator (described in Section 3.3.3.1) is designed to ignore equations with large violations of diagonal dominance.

3.4.2 Thermal flow

We now consider the energy conservation equation and temperature unknown separately from the saturation/concentration unknowns. We rewrite the linear system (3.8) as

$$Ax = \begin{bmatrix} A_{pp} & A_{pT} & A_{ps} \\ A_{Tp} & A_{TT} & A_{Ts} \\ A_{sp} & A_{sT} & A_{ss} \end{bmatrix} \begin{bmatrix} x_p \\ x_T \\ x_s \end{bmatrix} = \begin{bmatrix} b_p \\ b_T \\ b_s \end{bmatrix} = b, \tag{3.59}$$

where $A_{TT} \in \mathbb{R}^{N_p \times N_p}$ represents the temperature block coefficients, and $A_{Tp} \in \mathbb{R}^{N_p \times N_p}$, $A_{pT} \in \mathbb{R}^{N_p \times N_p}$, $A_{Ts} \in \mathbb{R}^{N_p \times N_s}$, and $A_{sT} \in \mathbb{R}^{N_s \times N_p}$ represent the respective coupling coefficients. We denote the temperature solution and residual, x_T and b_T, respectively.

We reproduce and comment on the analysis done in [42] for thermal simulation. Recall that U_α and H_α are the internal energy and enthalpy of phase α, respectively,

H_r is the enthalpy of the rock, k_T is the thermal conductivity, and T is the temperature. The discretized version of (2.5) in cell i and time-step n is

$$\begin{aligned}[A_{\text{energy}}]_i^n :=& \frac{1}{\Delta t}\left(\left[\sum_{\alpha=1}^{n_{\text{phases}}} \phi\rho_\alpha S_\alpha U_\alpha + (1-\phi)\rho_r H_r\right]_i^n \right.\\ &\left. - \left[\sum_{\alpha=1}^{n_{\text{phases}}} \phi\rho_\alpha S_\alpha U_\alpha + (1-\phi)\rho_r H_r\right]_i^{n-1}\right)\\ &- \frac{1}{hA}\sum_{\alpha=1}^{n_{\text{phases}}}\left([T_\alpha H_\alpha]_{i-\frac{1}{2}}^n([p_\alpha]_{i-1}^n - [p_\alpha]_i^n) + [T_\alpha H_\alpha]_{i+\frac{1}{2}}^n([p_\alpha]_{i+1}^n - [p_\alpha]_i^n)\right)\\ &- \frac{1}{hA}\left([k_T]_{i-1/2}^n([T]_{i-1}^n - [T]_i^n) + [k_T]_{i+1/2}^n([T]_{i+1}^n - [T]_i^n)\right), \end{aligned} \tag{3.60}$$

where we ignored gravity and well terms for simplicity. We recall that H_α and U_α are the phase enthalpies and internal energies of phase α, respectively, and that k_T represents the total thermal conductivity.

According to [42], the derivatives of density and transmissibility with respect to temperature are negative but are dominated by the positive derivatives of enthalpy and internal energy. Hence, in typical thermal simulation, A_{TT} is expected to be an M-matrix. Diagonal dominance depends on how the advection and accumulation dominate the diffusion. Similarly, because of the negative derivatives, $-A_{pT}$ and the blocks of $-A_{sT}$ are expected to be M-matrices according to [42].

However, our investigation reveals that temperature derivatives of the transmissibility are actually positive, at least for the models described in Chapter 2. Indeed, the influence of viscosity dominates density for the wide majority of relevant parameter regimes for our models. In the case of A_{TT}, the positive influence of enthalpy implies that it remains an M-matrix. On the other hand, $-A_{pT}$ and the blocks of $-A_{sT}$ are no longer expected to be M-matrices since the influence of the transmissibility dominates over the accumulation term, except for very small time-steps.

The A_{Tp} block will have similar properties to A_{pp} and the A_{sp} block, i.e. M-matrix properties.

Discounting the heat diffusion term and heat of rock, the energy conservation equation (3.60) is quite similar to an enthalpy-weighted sum of mass conservation equations of the phases (3.57). Since the discounted terms have little dependence on saturations/concentrations, the A_{Ts} blocks should have similar properties to the diagonal blocks of A_{ss} according to [42]. However, this does not account for unknowns eliminated by constraints. As discussed in Section 3.4.3, the contribution will be

negative in cells where eliminated saturations/concentrations dominate. Rows in A_{Ts} may therefore alternate between positive and negative entries.

3.4.3 Compositional flow

Similarly, we can define the discretized version of the mass conservation equation (2.29) for component c as follows:

$$\begin{aligned}[A_c]_i^n \coloneqq & \frac{1}{\Delta t} \sum_{\alpha=1}^{n_{\text{phases}}} \left([X_{\alpha,c}\phi\rho_\alpha S_\alpha]_i^n - [X_{\alpha,c}\phi\rho_\alpha S_\alpha]_i^{n-1} \right) \\ & - \frac{1}{hA} \sum_{\alpha=1}^{n_{\text{phases}}} \left([X_{\alpha,c}T_\alpha]_{i-\frac{1}{2}}^n ([p_\alpha]_{i-1}^n - [p_\alpha]_i^n) + [X_{\alpha,c}T_\alpha]_{i+\frac{1}{2}}^n ([p_\alpha]_{i+1}^n - [p_\alpha]_i^n) \right) \\ & - \sum_{\alpha=1}^{n_{\text{phases}}} [X_{\alpha,c}\rho_\alpha q_\alpha]_i^n, \end{aligned} \tag{3.61}$$

where we have ignored gravity for simplicity. Recall $X_{\alpha,c}$ is the concentration of component c within phase α.

The analysis done by Gries for the blocks with pressure derivatives easily extends in this case. The only difference with the multiphase case is that we need to sum over the different phases. Since the summands have similar properties with respect to pressure, the pressure block has similar properties to the multiphase case.

However, it is simply assumed in [42] that the properties remain the same for the blocks with saturation/concentration derivatives. We will see below that the properties of the blocks greatly depend on the alignment and choice of unknowns.

To illustrate the properties of compositional flow models, we consider a model and unknown selection similar to the one in [42]. We consider a model with a water component, which resides in an aqueous phase only, and three hydrocarbon components which can reside in a gas or oil phase. Since the water component resides in a single phase, its value is one and so its mass conservation equation is the same as in Section 3.4.1. We set $S_o = 1 - S_g - S_w$, and reduce all concentrations using constraints and equilibrium relations apart from one, say $X_{g,3}$. The system consists of four mass conservation equations of the form (2.29) (water and hydrocarbon components 1, 2, and 3) and four unknowns (pressure p, saturations S_w and S_g, and a concentration $X_{g,3}$). We chose three hydrocarbon components since it is the smallest number of components for which a concentration is needed explicitly.

It is also unclear how the alignment of equations and unknowns should be done. We choose the following alignment:

$$A_{pp} = \left[\frac{\partial [A_1]_i^n}{\partial [p]_i^n}\right], \qquad A_{ps} = \left[\frac{\partial [A_1]_i^n}{\partial [S_g]_i^n} \quad \frac{\partial [A_1]_i^n}{\partial [X_{g,3}]_i^n} \quad \frac{\partial [A_1]_i^n}{\partial [S_w]_i^n}\right],$$

$$A_{sp} = \begin{bmatrix} \frac{\partial [A_2]_i^n}{\partial [p]_i^n} \\ \frac{\partial [A_3]_i^n}{\partial [p]_i^n} \\ \frac{\partial [A_w]_i^n}{\partial [p]_i^n} \end{bmatrix}, \qquad A_{ss} = \begin{bmatrix} \frac{\partial [A_2]_i^n}{\partial [S_g]_i^n} & \frac{\partial [A_2]_i^n}{\partial [X_{g,3}]_i^n} & \frac{\partial [A_2]_i^n}{\partial [S_w]_i^n} \\ \frac{\partial [A_3]_i^n}{\partial [S_g]_i^n} & \frac{\partial [A_3]_i^n}{\partial [X_{g,3}]_i^n} & \frac{\partial [A_3]_i^n}{\partial [S_w]_i^n} \\ \frac{\partial [A_w]_i^n}{\partial [S_g]_i^n} & \frac{\partial [A_w]_i^n}{\partial [X_{g,3}]_i^n} & \frac{\partial [A_w]_i^n}{\partial [S_w]_i^n} \end{bmatrix}. \tag{3.62}$$

The properties of the blocks are different from those of (3.58) because of the multiple eliminations done with saturation and concentration constraints. The saturation-related blocks of (3.58) are either highly diagonally dominant M-matrices or the negative equivalent. This sign difference comes from using the saturation constraint to eliminate one saturation unknown. In a similar manner, we want to determine the signs of the diagonals of the blocks of A_{ps} and A_{ss} in (3.62).

Similarly to the diagonal blocks of A_{ss} in (3.58), A_{44} has a positive diagonal. Since S_w only appears in the other equations through $-S_o$, the A_{14}, A_{24} and A_{34} have negative diagonals. In a similar way, the A_{33} block has a positive diagonal. The A_{13} and A_{23} blocks have negative diagonals since $X_{g,1} = 1 - X_{g,3}$ and $X_{g,2} = 1 - X_{g,3}$. The A_{43} block should be null since $X_{g,3}$ does not appear in the water equation.

For the A_{12}, A_{22}, and A_{32} blocks, S_g appears explicitly but also through $-S_o$. This means that the sign of the diagonal terms depends on which phase dominates in a certain cell. The properties of the A_{42} block depend on how the relative permeability depends on S_g, which should result in a non-negative diagonal according to [42]. In practice, they can also be negative. We illustrate the sign of the diagonals in Table 3.1, where the rows represent the mass conservation equations, and the columns, the unknowns. The unpredictable sign pattern in compositional cases may prohibit the use of methods initially developed for the more predictable multiphase problems.

Table 3.1: Signs of the diagonals of the blocks in (3.62).

	S_g	$X_{g,3}$	S_w
hydrocarbon 1	+/-	-	-
hydrocarbon 2	+/-	-	-
hydrocarbon 3	+/-	+	-
water	?	0	+

3.4.4 Decoupling operators

We investigate the properties of the approximately decoupled pressure block $\tilde{A}_{pp} = A_{pp} - MA_{sp}$, where M is a block-diagonal matrix. For a problem with k non-pressure unknowns, elements of $\tilde{A}_{pp}$ are given at the cell level by

$$\tilde{A}_{pp}^{i,j} = A_{pp}^{i,j} - \begin{pmatrix} M_1^i & \cdots & M_k^i \end{pmatrix} A_{sp}^{i,j}. \tag{3.63}$$

The decoupling coefficients are given by $M^i = A_{ps}^{i,i}(A_{ss}^{i.i})^{-1}$, $M^i = M_{ps}^{i,i}(M_{ss}^{i,i})^{-1}$, and $M^i = (A_{ps}^{i,*}A_{ss}^{i,*\top})(A_{ss}^{i,*}A_{ss}^{i,*\top})^{-1}$ for QI, TI, and LSQ, respectively.

The decoupling coefficients M_x^i, $x = 1, \ldots, k$, can vary greatly for different values of i, which can result in a great loss of symmetry in $\tilde{A}_{pp}$. Symmetry is a desirable property for AMG, but not necessary for convergence in most cases. More important for AMG than symmetry is that $\tilde{A}_{pp}$ is not indefinite [97]. Let us assume that A_{pp} and the blocks of A_{sp} are M-matrices. Then, if the decoupling coefficients M_x^i are negative, we easily get from (3.63) that $\tilde{A}_{pp}$ is also an M-matrix. If some decoupling coefficients are positive, this is no longer guaranteed.

3.4.4.1 Multiphase flow

The properties of $\tilde{A}_{pp}$ are somewhat predictable if we consider multiphase flow as in Section 3.4.1. Some details are proven in Appendix B for the case where $A_{ss}^{i,i}$ is 2×2 (the same as in Section 3.4.3). We will refer to the Appendix when using the results therein.

The off-diagonals of $M_{ss}^{i,i}$ represent the derivatives of the mass accumulation term with respect to the other saturations. Since ρ_β only depends on p_β and ϕ is usually constant with respect to S_γ, these off-diagonals are null. Since the entries of $M_{ps}^{i,i}$ are expected to be negative, we easily get that the decoupling coefficients M_x^i are negative for TI.

In the case of QI, the off-diagonals of $A_{ss}^{i,i}$ come only from the flux terms. As mentioned in Section 3.4.1, these are expected to be non-negative in [42], but are often observed in practice to be negative. We also have that the entries of $A_{ps}^{i,i}$ are expected to be negative. If we suppose that the off-diagonal entries are non-negative, a high enough diagonal dominance is sufficient for the decoupling coefficients M_x^i to be negative for QI (see Appendix B). If the off-diagonal entries are negative, then $A_{ss}^{i,i}$ being an M-matrix is sufficient for the decoupling coefficients to be negative (see Appendix B). In practice, the coefficients for QI are usually negative, but there is no guarantee.

For LSQ, diagonal dominance is sufficient for the decoupling coefficients to be negative in the case where the off-diagonal entries are non-negative (see Appendix B). In other cases, it is difficult to interpret explicit conditions for the coefficients to be negative (given in Appendix B). In this case, $A_{ss}^{i,i}$ being an M-matrix is not sufficient for the decoupling coefficients to be negative.

In conclusion, TI is robust for multiphase flow in the sense that $\tilde{A}_{pp}$ is an M-matrix if A_{pp} and the blocks of A_{sp} are. For QI and even more so for LSQ, $\tilde{A}_{pp}$ is an M-matrix only under certain conditions. However, these conditions are usually fulfilled, at least in the observed cases. In the presence of injection wells, A_{pp} and the blocks of A_{sp} may no longer be M-matrices and so these observations no longer hold for all cells in $\tilde{A}_{pp}$.

3.4.4.2 Thermal flow

In commercial reservoir simulators, the energy and temperature blocks are included in the decoupling strategy. As discussed in Section 3.4.2, the properties of the additional thermal blocks are different.

Consider a thermal version of the pressure block (3.63), where the energy equation is part the secondary equations. Assuming that A_{Tp} is also an M-matrix, we require that the decoupling coefficients be negative. To use the results from Appendix B, we can consider a two-phase case so that $A_{ss}^{i,i}$ (including temperature) is a 2×2 matrix.

For TI, we no longer have that $M_{ss}^{i,i}$ is a diagonal block since $M_{sT}^{i,i}$ and $M_{Ts}^{i,i}$ are not null. $M_{sT}^{i,i}$ are expected to be negative due to the positive dependence of density. The sign of $M_{Ts}^{i,i}$ will vary depending on which phase dominates in cell i. Both $M_{ps}^{i,i}$ and $M_{pT}^{i,i}$ should be negative. It is unclear if $M_{ss}^{i,i}$ satisfies the properties in Appendix B to result in negative coefficients. For our test cases, however, we have observed that TI results in negative coefficients.

For QI, as discussed in Section 3.4.2, the main difficulty lies in $A_{pT}^{i,i}$ usually being positive. Therefore, it is even less clear when QI results in negative coefficients. We observed thermal cases where QI fails due to positive decoupling coefficients, while TI performs well. The same difficulty arises for LSQ.

The superiority of TI over QI and LSQ is even clearer here than in Section 3.4.4.1. It is not as clear, however, if TI is robust in the sense that $\tilde{A}_{pp}$ is an M-matrix if the blocks of A_{sp} and A_{Tp} are.

3.4.4.3 Compositional flow

Properties of $\tilde{A}_{pp}$ are unpredictable in compositional cases such as the one described in Section 3.4.3.

For a model with water and three hydrocarbon components, $A_{ss}^{i,i}$ is a 3×3 matrix. A sufficient condition for the decoupling coefficients to be negative for TI (QI) is that $M_{ss}^{i,i}$ ($A_{ss}^{i,i}$) be monotone and $M_{ps}^{i,i}$ ($A_{ps}^{i,i}$) have negative entries. Looking at Table 3.1, this is hardly guaranteed for any equation/unknown alignment. In the few observed compositional cases, the decoupling coefficients are often positive (sometimes more than half of them). This puts into question the use of decoupling operators in compositional cases. In contrast, the DRS and FRS decoupling operators always result in negative coefficients.

In practice, if a diagonal entry in $\tilde{A}_{pp}$ is negative, the corresponding row is multiplied by -1 before the AMG step of CPR. Although M-matrix properties are not guaranteed, this often improves the performance of AMG. This multiplication is also used to counter the negative influence of injection wells.

Overall, the properties of the linearized systems obtained for compositional models are different from those obtained from multiphase models. This puts into question the use of traditional decoupling operators concerning the applicability of AMG for the pressure system. On the other hand, ignoring the couplings in the pressure equation might ruin the effectiveness of CPR. Therefore, the success of CPR for compositional cases appears to be dependent on the effectiveness of AMG for problems which stray from ellipticity.

Chapter 4

Spatial Discretization

Finite Volume methods [56] are commonly used in reservoir simulation, as briefly described at the beginning of Section 3.1. Since the flux entering a given volume is identical to that leaving an adjacent one, these methods are conservative. Additionally, upwind schemes introduce substantial numerical diffusion, which helps with stability.

The standard flux approximation is a first-order approximation called the two-point flux approximation (TPFA), described later in this Chapter. Multipoint flux approximations (MPFA) [1] can lead to higher-order approximations and convergence on more unstructured grids. However, they may violate a maximum principle (and lead to spurious oscillations).

Discontinuous Galerkin (DG) methods [80] can also be used for Darcy flow problems. When an accurate solution of the velocity is required, mixed formulations are often preferred [15]. Similarly, stabilized continuous Finite Element methods for mixed formulations have been proposed [64]. Since linearized systems resulting from mixed formulations include velocity, solution techniques will differ from the ones considered in this thesis.

To have easy access to the different preconditioners and solvers provided by the PETSc library [9], we want to use the open source Finite Element software Firedrake [77] to handle spatial discretization. The user must express the weak form of their PDE system in the Unified Form Language (UFL) [5] from the FEniCS project [4, 59]. Therefore, we require a weak formulation of a Finite Volume method.

In this chapter, we present a piecewise constant discontinuous Galerkin method (DG(0)) that is equivalent to the Finite Volume method used in reservoir simulation and is based on the description in [79]. The resulting weak formulation allows us to implement our problem in Firedrake, and could similarly be used with FEniCS.

4.1 DG(0) discretization examples

We present a DG(0) method for simple examples before applying it to our problems of interest in Sections 4.3 and 4.4.

Let $\mathcal{T} = \{E_i, i \in \mathcal{I}\}$ be a partition of a domain Ω into open element domains E_i such that the union of their closure is $\overline{\Omega}$, where $\mathcal{I}$ is a set of indices. Let the interior facet $e_{ij} = \overline{E}_i \cap \overline{E}_j$ and let Γ_{int} denote the union of all interior facets. Let connection set $\mathcal{N}(i)$ denote the set of indices j such that $|e_{ij}| > 0$.

4.1.1 Heat equation

We begin by presenting a DG(0) scheme for the heat equation

$$\frac{\partial u}{\partial t} - \nabla^2 u = 0 \text{ in } \Omega, \tag{4.1}$$

$$u = f \text{ on } \Gamma_D, \quad \nabla u \cdot \mathbf{n} = g \text{ on } \Gamma_N. \tag{4.2}$$

The variational problem for (4.1)-(4.2) on a single cell E_i is: find u such that

$$\int_{E_i} \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_{E_i} \nabla u \cdot \nabla v \,\mathrm{d}x - \int_{\partial E_i} v \nabla u \cdot \mathbf{n}_i \,\mathrm{d}s = 0 \quad \text{for all test functions } v, \tag{4.3}$$

where $\mathbf{n}_i$ is the outward normal to E_i. Let us first consider E_i such that $\partial E_i \in \Gamma_{\text{int}}$. Let h_i denote the center point of cell E_i. For the flux on the interior facets, we choose the following linear approximation:

$$\int_{\partial E_i} v \nabla u \cdot \mathbf{n}_i \,\mathrm{d}s \approx \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} v \,|_{E_i}^{e_{ij}} \frac{u \,|_{E_j}^{e_{ij}} - u \,|_{E_i}^{e_{ij}}}{\|h_j - h_i\|} \,\mathrm{d}s. \tag{4.4}$$

Here, $u \,|_{E_i}^{e_{ij}}$ denotes the limit of u in cell E_i as it goes to the edge e_{ij}.

We consider a piecewise constant approximation of our solution, i.e. in the approximation space $\mathcal{V}_h = \mathbb{P}^0_{\text{DG}}$ with basis $\{\phi_i = \mathbb{1}_{E_i} \mid i \in \mathcal{I}\}$. The DG(0) approximation of the solution u is $u_h = \sum_{i \in \mathcal{I}} u_i \phi_i$. For this approximation, on E_i, $v_h \in \mathcal{V}_h$ is constant and $\nabla v_h = 0$. Therefore, (4.3) becomes the following equation for the coefficients:

$$\int_{E_i} \frac{\partial u_i}{\partial t} v_i \,\mathrm{d}x - \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} v_i \frac{u_j - u_i}{\|h_j - h_i\|} \,\mathrm{d}s = 0, \tag{4.5}$$

for all $v_h \in \mathcal{V}_h$ such that $v_h = \sum_{i \in \mathcal{I}} v_i \phi_i$. Equivalently, we can replace the test functions by the basis functions. Note that $\int_{E_i} \mathbb{1}_{E_i} \,\mathrm{d}x = |E_i|$ is the volume of cell E_i, and $\int_{e_{ij}} \mathbb{1}_{E_i} \,\mathrm{d}s = |e_{ij}|$ is the area of facet e_{ij}. The equation for $\mathbb{1}_{E_i}$ is thus

$$\frac{\partial u_i}{\partial t} |E_i| - \sum_{j \in \mathcal{N}(i)} \frac{u_j - u_i}{\|h_j - h_i\|} |e_{ij}| = 0, \tag{4.6}$$

which is a Finite Volume approximation of the heat equation. In reservoir simulation, this way of approximating the interior facet integrals is known as a "two-point flux" approximation (TPFA). A representation of TPFA is illustrated in Figure 4.1. In order for such a Finite Volume method to converge, the grid must satisfy a certain orthogonality property [35]. In brief, in each cell, there exists a point called the center of the cell such that for any adjacent cell, the straight line between the two centers is orthogonal to the boundary between the cells. For the examples in this thesis, we choose Cartesian quadrilateral meshes, which easily satisfy this condition.

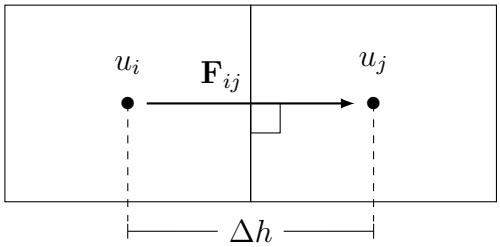


Figure 4.1: Two-point flux approximation for flux $\mathbf{F}_{ij}$ between adjacent cells i and j between distance Δh between cell centers.

If instead E_i is a boundary element, then the boundary integral becomes

$$\int_{\partial E_i} v \nabla u \cdot \mathbf{n} \, \mathrm{d}s = \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} v \nabla u \cdot \mathbf{n}_i + \int_{\partial E_i \cap \Gamma_N} v \nabla u \cdot \mathbf{n} \, \mathrm{d}s, \tag{4.7}$$

and we enforce the Dirichlet boundary condition geometrically by restricting $u = f$ on the cells adjacent to Γ_D. We again use the two-point flux approximation

$$\int_{\partial E_i} v \nabla u \cdot \mathbf{n}_i \, \mathrm{d}s \approx \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} v_i \frac{u_j - u_i}{\|h_j - h_i\|} \, \mathrm{d}s + \int_{\partial E_i \cap \Gamma_N} v_i g \, \mathrm{d}s. \tag{4.8}$$

For each $i \in \mathcal{I}$, we have

$$\int_{E_i} \frac{\partial u_i}{\partial t} v_i \, \mathrm{d}x - \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} v_i \frac{u_j - u_i}{\|h_j - h_i\|} \, \mathrm{d}s - \int_{\partial E_i \cap \Gamma_N} v_i g \, \mathrm{d}s = 0. \tag{4.9}$$

We now want to express the problem over the whole mesh by summing over all $i \in \mathcal{I}$. We make a small abuse of notation by denoting the cell integrals by $\sum_{i \in \mathcal{I}} \int_{E_i} \cdot \, \mathrm{d}x = \int_\Omega \cdot \, \mathrm{d}x$. Similarly, we denote the facet integrals by $\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{N}(i)} \int_{e_{ij}} \cdot \, \mathrm{d}s = \int_{\Gamma_{\text{int}}} \cdot \, \mathrm{d}S$, noting that each interior facet is visited twice. To be consistent with UFL notation, the upper and lower case $\mathrm{d}S$ and $\mathrm{d}s$ refer to the integrals over the interior skeleton, and boundary, respectively.

For a given ordering of the indices in $\mathcal{I}$, we denote by u^+ and u^- the limit values of u for two cells sharing an edge (which in our case are just the constant values in those cells). Summing (4.9) over all $i \in \mathcal{I}$, we obtain

$$\int_\Omega \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_{\Gamma_{\text{int}}} (v^+ - v^-) \frac{u^+ - u^-}{\|h^+ - h^-\|} \,\mathrm{d}S - \int_{\Gamma_N} vg \,\mathrm{d}s = 0. \tag{4.10}$$

We define the jump of v as $[v] = v^+ - v^-$. We then get the following problem: find $u \in \mathbb{P}^0_{\text{DG}}$ such that

$$\int_\Omega \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_{\Gamma_{\text{int}}} [v] \frac{[u]}{\|h^+ - h^-\|} \,\mathrm{d}S - \int_{\Gamma_N} vg \,\mathrm{d}s = 0, \tag{4.11}$$

for all $v \in \mathbb{P}^0_{\text{DG}}$.

4.1.2 Upwinding

We now consider an upwind Godunov method [40] for the advection equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (u\mathbf{w}) = 0 \text{ in } \Omega, \tag{4.12}$$

$$u = f \text{ on } \Gamma_D, \quad u\mathbf{w} \cdot \mathbf{n} = g \text{ on } \Gamma_N, \tag{4.13}$$

where $\mathbf{w}$ is a given vector field. For an interior E_i, the upwind scheme is given by

$$\int_{E_i} \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_{\partial E_i} v u^{\text{up}} \mathbf{w} \cdot \mathbf{n}_i \,\mathrm{d}s = 0, \tag{4.14}$$

where u^{up} is the upwind value of u, which, for a facet e shared by E_1 and E_2 and $\mathbf{n}_e$ pointing from E_1 to E_2, is given by

$$u^{\text{up}} = \begin{cases} u|^e_{E_1} & \text{if } \mathbf{w} \cdot \mathbf{n}_e \geq 0, \\ u|^e_{E_2} & \text{if } \mathbf{w} \cdot \mathbf{n}_e < 0. \end{cases} \tag{4.15}$$

For the full semi-discretized problem we have: find $u \in \mathbb{P}^0_{\text{DG}}$ such that

$$\int_\Omega \frac{\partial u}{\partial t} v \,\mathrm{d}x + \int_{\Gamma_{\text{int}}} [v] u^{\text{up}} \mathbf{w} \cdot \mathbf{n}_e \,\mathrm{d}S \,\mathrm{d}s - \int_{\Gamma_N} vg \,\mathrm{d}s = 0, \tag{4.16}$$

for all $v \in \mathbb{P}^0_{\text{DG}}$. Again, the Dirichlet boundary condition is enforced geometrically. Here, $\mathbf{n}_e$ denotes the unit outward pointing normal for a currently visited cell (again, each side of the facet is visited).

4.1.3 Heterogeneous coefficients

In reservoir simulation, the permeability tensors are often highly heterogeneous and piecewise constant (on cells). For two-point flux approximation, it is standard to approximate the permeability on the facets using a harmonic average. This is derived by considering piecewise constant coefficients [35, 73].

Consider the following diffusion problem:

$$\nabla \cdot K\nabla u = 0 \text{ in } \Omega, \qquad K\nabla u \cdot \mathbf{n} = 0, \text{ on } \partial\Omega \tag{4.17}$$

where K is a scalar field which is piecewise constant on a given mesh. The harmonic average of K on a facet is given by

$$\{\!\{K\}\!\} = \frac{2K^+K^-}{K^+ + K^-}. \tag{4.18}$$

The DG(0) problem for (4.17) is: find $u \in \mathbb{P}^0_{\mathrm{DG}}$ such that

$$\int_{\Gamma_{\mathrm{int}}} \{\!\{K\}\!\}[v]\frac{[u]}{\|h^+ - h^-\|}\,\mathrm{d}S = 0, \tag{4.19}$$

for all $v \in \mathbb{P}^0_{\mathrm{DG}}$.

In reservoir simulation, the permeability can also be anisotropic. It is then given by a diagonal tensor field of the form $\mathbf{K} = \mathrm{diag}(K_x, K_y, K_z)$. The harmonic average of the tensor field $\mathbf{K}$ on a facet is given by $\{\!\{\mathbf{K}\}\!\} = \mathrm{diag}(\{\!\{K_x\}\!\}, \{\!\{K_y\}\!\}, \{\!\{K_z\}\!\})$.

4.1.4 Darcy flow

Consider the following Darcy flow problem:

$$-\nabla \cdot \left(\mathbf{K}\frac{\rho}{\mu}(\nabla p - \rho\mathbf{g})\right) = 0 \quad \text{in } \Omega, \tag{4.20}$$

where ρ and μ are scalar fields (perhaps functions of p). It is standard in commercial reservoir simulators to use the upwind values for the transported quantities (ρ/μ), and the average value for the density ρ which is multiplied to the gravitational acceleration $\mathbf{g}$. The DG(0) formulation for the equation above is: find $p \in \mathbb{P}^0_{\mathrm{DG}}$ such that

$$\int_{\Gamma_{\mathrm{int}}} [q]\left(\{\!\{\mathbf{K}\}\!\}\frac{\rho^{\mathrm{up}}}{\mu^{\mathrm{up}}}\left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho\}\mathbf{g}\cdot\mathbf{n}_e\right)\right)\mathrm{d}S = 0, \tag{4.21}$$

for all $q \in \mathbb{P}^0_{\mathrm{DG}}$. Here the brackets $\{\}$ denote the average across the facets such that $\{\rho\} = (\rho^+ + \rho^-)/2$. Since the velocity is given by Darcy's law, the upwind quantities are given by

$$(u)^{\mathrm{up}} = \begin{cases} u\,|^e_{E_1} & \text{if } \dfrac{[p]}{\|h^+ - h^-\|} - \{\rho\}\mathbf{g}\cdot\mathbf{n}_e \geq 0, \\ u\,|^e_{E_2} & \text{if } \dfrac{[p]}{\|h^+ - h^-\|} - \{\rho\}\mathbf{g}\cdot\mathbf{n}_e < 0. \end{cases} \tag{4.22}$$

4.2 Numerical tests

Even though the Finite Volume method used here is standard, formulating it as a variational problem is not. Therefore, we perform some numerical tests to assess the convergence of our discretization scheme. We will manufacture analytical solutions. In each case, we set a Dirichlet boundary condition equal to the analytical solution.

4.2.1 Heterogeneous Poisson problem

Consider the following problem for u:

$$-\nabla \cdot K\nabla u = f \text{ in } [0,1] \times [0,1]. \tag{4.23}$$

For $K = 1 + x + y$, the analytical solution $u = 1 + x^2 + y^2$ returns the right-hand side $f = -8x - 10y - 6$.

We consider a $N \times N$ grid and the resulting DG(0) discretization similar to (4.19). All quantities are evaluated at the cell centers. We evaluate the relative L^2 error of the DG(0) solution as we refine the mesh. In Table 4.1, we observe that the method converges at first order.

Table 4.1: Relative L^2 error and observed convergence rate for the heterogeneous Poisson problem on $N \times N$ grid.

N	4	8	16	32	64	128	256	512
L^2 error	0.0882	0.0442	0.0221	0.0110	0.00552	0.00276	0.00138	6.91e-4
rate	-	0.997	0.999	1	1	1	1	1

4.2.2 Advection problem

Consider the following advection problem for u with velocity field $\mathbf{K}\nabla p$:

$$-\nabla \cdot (u\mathbf{K}\nabla p) = f \text{ in } [0,1] \times [0,1], \tag{4.24}$$

For $K_x = 1 + x$, $K_y = 1 + x + y$, $p = 1 + x + y$, the analytical solution $u = 1 + x^2 + 2y^2$ returns the right-hand side $f = -4x^2 - 8y^2 - 2x - 4y - 2$.

We consider an $N \times N$ grid and the resulting DG(0) discretization similar to (4.21), i.e. using upwinding for u and TPFA for ∇p. Again, all quantities are evaluated at the cell centers. In Table 4.2, we observe that the method eventually reaches first order convergence. For this toy problem, upwinding is not necessary and actually results in larger errors than using averaging for u.

Table 4.2: Relative L^2 error and observed convergence rate for the advection problem on $N \times N$ grid.

N	4	8	16	32	64	128	256	512
L^2 error	0.107	0.0834	0.0555	0.0323	0.0175	0.00916	0.00469	0.00237
order	-	0.361	0.589	0.778	0.882	0.938	0.967	0.983

4.2.3 Advection-diffusion problem

Consider the following advection-diffusion problem for u with velocity field $-\mathbf{K}\nabla p$:

$$-\nabla \cdot (u\mathbf{K}\nabla p) - \nabla \cdot \kappa \nabla u = f \text{ in } [0,1] \times [0,1], \tag{4.25}$$

For $K_x = 1 + x$, $K_y = 1 + x + y$, $p = 1 + x + y$, $\kappa = 0.5(1 + x + y)$, the analytical solution $u = 1 + x^2 + 2y^2$ returns the right-hand side $f = -4x^2 - 8y^2 - 6x - 9y - 5$.

Again, we consider an $N \times N$ grid and the resulting DG(0) discretization similar to (4.21), i.e. using upwinding for u and TPFA for both ∇p and ∇u. In Table 4.3, we observe that the method converges at first order in the L^2 norm.

Table 4.3: Relative L^2 error and observed convergence rate for the advection-diffusion problem on $N \times N$ grid.

N	4	8	16	32	64	128	256	512
L^2 error	0.0883	0.0449	0.0228	0.0115	0.00581	0.00292	0.00146	0.000731
order	-	0.976	0.975	0.983	0.990	0.995	0.997	0.999

4.3 Single-phase flow

Without assuming any regularity for the solution space, the weak form of equations (2.3), (2.12) on an interior cell E_i is given by

$$\int_{E_i} \phi \frac{\partial \rho}{\partial t} q \,\mathrm{d}x + \int_{E_i} \left(\rho \frac{\mathbf{K}}{\mu} (\nabla p - \rho \mathbf{g}) \right) \cdot \nabla q \,\mathrm{d}x - \int_{\partial E_i} q \left(\rho \frac{\mathbf{K}}{\mu} (\nabla p - \rho \mathbf{g}) \right) \cdot \mathbf{n}_i \,\mathrm{d}s = \int_{E_i} f q \,\mathrm{d}x, \tag{4.26}$$

$$\int_{E_i} \phi \frac{\partial}{\partial t} (\rho c_v T) r \,\mathrm{d}x + \int_{E_i} (1-\phi) \frac{\partial}{\partial t} (\rho_r c_r T) r \,\mathrm{d}x + \int_{E_i} \left(\rho c_v T \frac{\mathbf{K}}{\mu} (\nabla p - \rho \mathbf{g}) \right) \cdot \nabla r \,\mathrm{d}x - \int_{\partial E_i} r \left(\rho c_v T \frac{\mathbf{K}}{\mu} (\nabla p - \rho \mathbf{g}) \right) \cdot \mathbf{n}_i \,\mathrm{d}s + \int_{E_i} (k_T \nabla T) \cdot \nabla r \,\mathrm{d}x - \int_{\partial E_i} r (k_T \nabla T) \cdot \mathbf{n}_i \,\mathrm{d}s = \int_{E_i} f_T r \,\mathrm{d}x, \tag{4.27}$$

for all test functions q, r.

4.3.1 Semidiscrete problem

We now discretize the system (2.3), (2.4), (2.12), (2.13) from Section 2.1.3 in space using the semidiscrete DG(0) formulation described above. Assuming homogeneous Neumann boundary conditions, the variational problem is: find the approximation $(p, T) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$ such that

$$\int_\Omega \phi \frac{\partial \rho}{\partial t} q \,\mathrm{d}x + \int_{\Gamma_{\mathrm{int}}} [q] \left(\{\!\{\mathbf{K}\}\!\} \frac{\rho^{\mathrm{up}}}{\mu^{\mathrm{up}}} \left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f q \,\mathrm{d}x = 0, \tag{4.28}$$

$$\begin{aligned} \int_\Omega \phi c_v \frac{\partial \rho T}{\partial t} r \,\mathrm{d}x + \int_\Omega (1-\phi) \rho_r c_r \frac{\partial T}{\partial t} r \,\mathrm{d}x \\ + \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{\mathbf{K}\}\!\} c_v \frac{\rho^{\mathrm{up}}}{\mu^{\mathrm{up}}} T^{\mathrm{up}} \left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho\} \mathbf{g} \cdot \mathbf{n}_e \right) \mathrm{d}S \\ + \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{k_T\}\!\} \frac{[T]}{\|h^+ - h^-\|} \,\mathrm{d}S - \int_\Omega f_T r \,\mathrm{d}x = 0, \end{aligned} \tag{4.29}$$

for all $(q, r) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$. Recall that the brackets $\{\cdot\}$ denote the average across the facets, and the double brackets $\{\!\{\cdot\}\!\}$ denote the harmonic average across the facets. The delta functions in the source/sink terms are approximated on coarse grids as

$$\delta(x) = \begin{cases} 1/|E_i| & \text{if } x \in E_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.30}$$

For mesh refinement cases, the volume of the approximated delta functions are kept constant so that they will be defined over several cells. In 2D, these cells will approximate a circle, and in 3D, a cylinder.

4.3.2 Fully discretized problem

For time discretization, we use the backward Euler method. We define the two following mappings, which are linear with respect with their last argument:

$$\begin{aligned} F_m(p^{n+1}, T^{n+1}; q) \coloneqq \int_\Omega \phi \frac{\rho^{n+1} - \rho^n}{\Delta t} q \,\mathrm{d}x \\ + \int_{\Gamma_{\mathrm{int}}} [q] \left(\{\!\{\mathbf{K}\}\!\} \frac{(\rho^{n+1})^{\mathrm{up}}}{(\mu^{n+1})^{\mathrm{up}}} \left(\frac{[p^{n+1}]}{\|h^+ - h^-\|} - \{\rho^{n+1}\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S \\ - \int_\Omega f^{n+1} q \,\mathrm{d}x, \end{aligned} \tag{4.31}$$

$$F_e(p^{n+1},T^{n+1};r) \coloneqq \int_\Omega \phi c_v \frac{\rho^{n+1}T^{n+1}-\rho^n T^n}{\Delta t} r\,\mathrm{d}x + \int_\Omega (1-\phi)\rho_r c_r \frac{T^{n+1}-T^n}{\Delta t} r\,\mathrm{d}x + \int_{\Gamma_{\text{int}}} [r]\{\!\{\mathbf{K}\}\!\} \frac{(\rho^{n+1})^{\text{up}}}{(\mu^{n+1})^{\text{up}}} (T^{n+1})^{\text{up}} \left(\frac{[p^{n+1}]}{\|h^+ - h^-\|} - \{\rho^{n+1}\}\mathbf{g}\cdot\mathbf{n}_e \right) \mathrm{d}S + \int_{\Gamma_{\text{int}}} [r]\{\!\{k_T\}\!\} \frac{[T^{n+1}]}{\|h^+ - h^-\|}\,\mathrm{d}S - \int_\Omega f_T^{n+1} r\,\mathrm{d}x. \quad (4.32)$$

Let $F(p,T;q,r) \coloneqq F_m(p,T;q) + F_e(p,T;r)$, which is linear in both q and r, but nonlinear in p and T. At each time-step, given the previous solution (p^n,T^n), we search for $(p^{n+1},T^{n+1}) \in \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}}$ such that

$$F(p^{n+1},T^{n+1};q,r) = 0 \quad \text{for all } (q,r) \in \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}}. \quad (4.33)$$

The mass and energy mappings, F_m and F_e, could also be scaled such that they are of similar size.

4.4 Multiphase flow

We describe the discretization for two-phase flow with an oil phase and a water phase, denoted o and w, respectively. Here, we set $S_w = 1 - S_o$. Without assuming any regularity for the solution space, the weak form of equations (2.18), (2.26) on an interior cell E_i is given by

$$\int_{E_i} \phi \frac{\partial(\rho_w(1-S_o))}{\partial t} q\,\mathrm{d}x + \int_{E_i} \left(\rho_w \frac{\mathbf{K}k_{rw}}{\mu_w}(\nabla p - \rho_w \mathbf{g}) \right) \cdot \nabla q\,\mathrm{d}x - \int_{\partial E_i} q \left(\rho_w \frac{\mathbf{K}k_{rw}}{\mu_w}(\nabla p - \rho_w \mathbf{g}) \right) \cdot \mathbf{n}_i\,\mathrm{d}s = \int_{E_i} f_w q\,\mathrm{d}x, \quad (4.34)$$

$$\int_{E_i} \phi \frac{\partial(c_{v,w}\rho_w(1-S_o)T)}{\partial t} r\,\mathrm{d}x + \int_{E_i} \phi \frac{\partial(c_{v,o}\rho_o S_o T)}{\partial t} r\,\mathrm{d}x + \int_{E_i} (1-\phi)\frac{\partial}{\partial t}(\rho_r c_r T) r\,\mathrm{d}x + \int_{E_i} \sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) \cdot \nabla r\,\mathrm{d}x - \int_{\partial E_i} r \sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha}(\nabla p - \rho_\alpha \mathbf{g}) \cdot \mathbf{n}_i\,\mathrm{d}s + \int_{E_i} (k_T \nabla T) \cdot \nabla r\,\mathrm{d}x - \int_{\partial E_i} r(k_T \nabla T) \cdot \mathbf{n}_i\,\mathrm{d}s = \int_{E_i} f_T r\,\mathrm{d}x, \quad (4.35)$$

$$\int_{E_i} \phi \frac{\partial(\rho_o S_o)}{\partial t} s\,\mathrm{d}x + \int_{E_i} \left(\rho_o \frac{\mathbf{K}k_{ro}}{\mu_o}(\nabla p - \rho_o \mathbf{g}) \right) \cdot \nabla s\,\mathrm{d}x - \int_{\partial E_i} s \left(\rho_o \frac{\mathbf{K}k_{ro}}{\mu_o}(\nabla p - \rho_o \mathbf{g}) \right) \cdot \mathbf{n}_i\,\mathrm{d}s = \int_{E_i} f_o s\,\mathrm{d}x, \quad (4.36)$$

for all test functions q, r, s.

4.4.1 Semidiscrete problem

We now discretize the system (2.18), (2.26), (2.19), (2.20), (2.27) from Section 2.2.3 in space using the semidiscrete DG(0) formulation described above. Assuming homogeneous Neumann boundary conditions, the variational problem is: find the approximation $(p, T, S_o) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$ satisfying the conservation of water mass equation:

$$\int_\Omega \phi \frac{\partial(\rho_w(1-S_o))}{\partial t} q \,\mathrm{d}x + \int_{\Gamma_{\mathrm{int}}} [q] \left(\{\!\{\mathbf{K}\}\!\} \frac{\rho_w^{\mathrm{up}} k_{rw}^{\mathrm{up}}}{\mu_w^{\mathrm{up}}} \left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho_w\}\mathbf{g}\cdot\mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f_w q \,\mathrm{d}x = 0, \quad (4.37)$$

the conservation of energy equation:

$$\int_\Omega \phi \frac{\partial(c_{v,w}\rho_w(1-S_o)T)}{\partial t} r \,\mathrm{d}x + \int_\Omega \phi \frac{\partial(c_{v,o}\rho_o S_o T)}{\partial t} r \,\mathrm{d}x + \int_\Omega (1-\phi)\rho_r c_r \frac{\partial T}{\partial t} r \,\mathrm{d}x + \sum_{\alpha=o,w} \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{\mathbf{K}\}\!\} k_{r\alpha}^{\mathrm{up}} c_{v,\alpha} \frac{\rho_\alpha^{\mathrm{up}}}{\mu_\alpha^{\mathrm{up}}} T^{\mathrm{up}} \left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho_\alpha\}\mathbf{g}\cdot\mathbf{n}_e \right) \mathrm{d}S + \int_{\Gamma_{\mathrm{int}}} [r] \{\!\{k_T\}\!\} \frac{[T]}{\|h^+ - h^-\|} \,\mathrm{d}S - \int_\Omega f_T r \,\mathrm{d}x = 0, \quad (4.38)$$

and the conservation of oil mass equation:

$$\int_\Omega \phi \frac{\partial(\rho_o S_o)}{\partial t} s \,\mathrm{d}x + \int_{\Gamma_{\mathrm{int}}} [s] \left(\{\!\{\mathbf{K}\}\!\} \frac{\rho_o^{\mathrm{up}} k_{ro}^{\mathrm{up}}}{\mu_o^{\mathrm{up}}} \left(\frac{[p]}{\|h^+ - h^-\|} - \{\rho_o\}\mathbf{g}\cdot\mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f_o s \,\mathrm{d}x = 0, \quad (4.39)$$

for all $(q, r, s) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$. The unit outward pointing normal of a cell is denoted by $\mathbf{n}_e$. The upwind quantities for phase α are given by

$$(u_\alpha)^{\mathrm{up}} = \begin{cases} u\,|^e_{E_1} & \text{if } \dfrac{[p]}{\|h^+ - h^-\|} - \{\rho_\alpha\}\mathbf{g}\cdot\mathbf{n}_e \geq 0, \\ u\,|^e_{E_2} & \text{if } \dfrac{[p]}{\|h^+ - h^-\|} - \{\rho_\alpha\}\mathbf{g}\cdot\mathbf{n}_e < 0. \end{cases} \quad (4.40)$$

The delta functions in the source terms are given by (4.30).

4.4.2 Fully discretized problem

For time discretization, we use the backward Euler method. We define three mappings, which are linear with respect with their last argument. As follows, we define

the conservation of water mass mapping:

$$F_w(p^{n+1}, T^{n+1}, S_o^{n+1}; q) \coloneqq \int_\Omega \phi \frac{\rho_w^{n+1}(1 - S_o^{n+1}) - \rho_w^n(1 - S_o^n)}{\Delta t} q \,\mathrm{d}x + \int_{\Gamma_{\text{int}}} [q] \left(\{\!\{\mathbf{K}\}\!\} \frac{(\rho_w^{n+1})^{\text{up}} (k_{rw}^{n+1})^{\text{up}}}{(\mu_w^{n+1})^{\text{up}}} \left(\frac{[p^{n+1}]}{\|h^+ - h^-\|} - \{\rho_w^{n+1}\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f_w^{n+1} q \,\mathrm{d}x, \quad (4.41)$$

the conservation of energy mapping:

$$\begin{aligned} F_e(p^{n+1}, T^{n+1}, S_o^{n+1}; r) \coloneqq & \int_\Omega \phi c_{v,w} \frac{(1 - S_o^{n+1}) \rho_w^{n+1} T^{n+1} - (1 - S_o^n) \rho_w^n T^n}{\Delta t} r \,\mathrm{d}x \\ & + \int_\Omega \phi c_{v,o} \frac{S_o^{n+1} \rho_o^{n+1} T^{n+1} - S_o^n \rho_o^n T^n}{\Delta t} r \,\mathrm{d}x + \int_\Omega (1 - \phi) \rho_r c_r \frac{T^{n+1} - T^n}{\Delta t} r \,\mathrm{d}x \\ & + \sum_{\alpha = o,w} \int_{\Gamma_{\text{int}}} [r] \{\!\{\mathbf{K}\}\!\} (k_{r\alpha}^{n+1})^{\text{up}} c_{v,\alpha} \frac{(\rho_\alpha^{n+1})^{\text{up}}}{(\mu_\alpha^{n+1})^{\text{up}}} (T^{n+1})^{\text{up}} \\ & \left(\frac{[p^{n+1}]}{\|h^+ - h^-\|} - \{\rho_\alpha^{n+1}\} \mathbf{g} \cdot \mathbf{n}_e \right) \mathrm{d}S \\ & + \int_{\Gamma_{\text{int}}} [r] \{\!\{k_T^{n+1}\}\!\} \frac{[T^{n+1}]}{\|h^+ - h^-\|} \,\mathrm{d}S - \int_\Omega f_T^{n+1} r, \end{aligned} \quad (4.42)$$

and the conservation of oil mass mapping:

$$F_o(p^{n+1}, T^{n+1}, S_o^{n+1}; s) \coloneqq \int_\Omega \phi \frac{\rho_o^{n+1} S_o^{n+1} - \rho_o^n S_o^n}{\Delta t} s \,\mathrm{d}x + \int_{\Gamma_{\text{int}}} [s] \left(\{\!\{\mathbf{K}\}\!\} \frac{(\rho_o^{n+1})^{\text{up}} (k_{ro}^{n+1})^{\text{up}}}{(\mu_o^{n+1})^{\text{up}}} \left(\frac{[p^{n+1}]}{\|h^+ - h^-\|} - \{\rho_o^{n+1}\} \mathbf{g} \cdot \mathbf{n}_e \right) \right) \mathrm{d}S - \int_\Omega f_o^{n+1} s \,\mathrm{d}x. \quad (4.43)$$

Let

$$F(p, T, S_o; q, r, s) \coloneqq F_w(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s), \quad (4.44)$$

which is linear in q, r, and s, but nonlinear in p, T, and S_o.

At each time-step, given the previous solution (p^n, T^n, S_o^n), we search for $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}}$ such that

$$F(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \quad \text{for all } (q, r, s) \in \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}} \times \mathbb{P}^0_{\text{DG}}. \quad (4.45)$$

In order to use the Schur complement approximation described later in Section 5.1.3.1, we instead need solve the equivalent problem: find $(p^{n+1}, T^{n+1}, S_o^{n+1}) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}$ such that

$$F^*(p^{n+1}, T^{n+1}, S_o^{n+1}; q, r, s) = 0 \quad \text{for all } (q, r, s) \in \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}} \times \mathbb{P}^0_{\mathrm{DG}}, \tag{4.46}$$

where

$$F^*(p, T, S_o; q, r, s) \coloneqq F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + F_o(p, T, S_o; s) \tag{4.47}$$

is F where F_w has been replaced by the following "pressure equation" mapping:

$$F_p(p, T, S_o; q) \coloneqq c_{v,w} F_w(p, T, S_o; q) + c_{v,o} F_o(p, T, S_o; q). \tag{4.48}$$

The choice of this weighted sum will be justified in Section 5.1.3.1. Recall that $c_{v,w}$ and $c_{v,o}$ are the specific heat coefficients of the water and oil phases, respectively.

Further scaling of the different equations might be needed for the robustness of the solution algorithms. We discuss this briefly at the end of Section 6.1.5.

Chapter 5

Preconditioning for Single-Phase Flow

We investigate preconditioning strategies for the single-phase thermal model given by (2.3), (2.4), (2.12), (2.13). For readability, we repeat the problem here: Find p, T such that

$$\phi\frac{\partial\rho}{\partial t} - \nabla\cdot\left(\rho\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g})\right) = f \quad \text{in } \mathbb{R}_+\times\Omega, \tag{5.1}$$

$$-\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g})\cdot\mathbf{n} = g_N \text{ on } \Gamma_N, \quad \text{and} \quad p = g_D \text{ on } \Gamma_D, \tag{5.2}$$

$$\phi\frac{\partial}{\partial t}(\rho c_v T) + (1-\phi)\frac{\partial}{\partial t}(\rho_r c_r T) - \nabla\cdot\left(\rho c_v T\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g})\right) - \nabla\cdot(k_T\nabla T) = f_T \quad \text{in } \mathbb{R}_+\times\Omega, \tag{5.3}$$

$$-\left(\rho c_v T\frac{\mathbf{K}}{\mu}(\nabla p - \rho\mathbf{g}) + k_T\nabla T\right)\cdot\mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \quad \text{and} \quad T = g_D^T \text{ on } \Gamma_D^T. \tag{5.4}$$

We will develop a new block preconditioner with an efficient Schur complement approximation in Sections 5.1.2 and 5.1.3. These results are also found in [82].

5.1 Preconditioning

The system of nonlinear equations (4.33) can be written as a system of nonlinear equations for the real coefficients p_i and T_i of the DG(0) functions p^{n+1} and T^{n+1}, respectively. Let x be the vector of these coefficients and $\mathcal{F}$ the function such that $\mathcal{F}(x) = 0$ is equivalent to (4.33). By linearizing this equation with Newton's method, we must solve at each iteration

$$\frac{\partial\mathcal{F}}{\partial x}\,|_{x=x_k}\,(x_{k+1} - x_k) = -\mathcal{F}(x_k). \tag{5.5}$$

The resulting linearized systems can be written as a block system of the form

$$A\delta x = \begin{bmatrix} A_{pp} & A_{pT} \\ A_{Tp} & A_{TT} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta T \end{bmatrix} = \begin{bmatrix} b_p \\ b_T \end{bmatrix} = b, \tag{5.6}$$

where $\delta x = x_{k+1} - x_k$ is the Newton increment. The different blocks are the discrete versions of Jacobian terms as follows

$$A_{pp} \sim \phi \frac{1}{\Delta t}(\rho)_p + \nabla \cdot (\rho \mathbf{u})_p - (f)_p, \tag{5.7}$$

$$A_{pT} \sim \phi \frac{1}{\Delta t}(\rho)_T + \nabla \cdot (\rho \mathbf{u})_T - (f)_T, \tag{5.8}$$

$$A_{Tp} \sim \phi \frac{1}{\Delta t}(\rho)_p c_v T + \nabla \cdot (c_v T(\rho \mathbf{u})_p) - (f_T)_p, \tag{5.9}$$

$$\begin{aligned} A_{TT} \sim \phi \frac{c_v(\rho + (\rho)_T T)}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla \cdot (c_v \rho \mathbf{u}) \\ + \nabla \cdot (c_v T(\rho \mathbf{u})_T) - \nabla \cdot (k_T \nabla) - (f_T)_T, \end{aligned} \tag{5.10}$$

where

$$(\rho \mathbf{u})_p = -\frac{\mathbf{K}}{\mu}\left(\rho(\nabla - (\rho)_p \mathbf{g}) + (\rho)_p(\nabla p - \rho \mathbf{g})\right), \tag{5.11}$$

and

$$(\rho \mathbf{u})_T = -\mathbf{K}\left[\left(\frac{\rho}{\mu}\right)_T (\nabla p - \rho \mathbf{g}) - \frac{\rho}{\mu}(\rho)_T \mathbf{g}\right]. \tag{5.12}$$

All coefficients in (5.7)-(5.12) are evaluated at the previous Newton iterate (p_k, T_k), and $(.)_p$ and $(.)_T$ denote the partial derivatives with respect to p and T, respectively.

The linearized system will be solved using GMRES (described in Section 3.2) preconditioned with the methods in this Section.

5.1.1 CPR for the single-phase case

In the case of multiphase flow in porous media, the standard preconditioner is the Constrained Pressure Residual method (CPR) [105], described in Section 3.3.3. Recall that CPR is a two-stage preconditioner where a restricted pressure system is solved in the first stage, followed by an approximate solution of the full system. We apply CPR to the single-phase case by considering temperature and the energy equation as secondary unknown and equation, respectively.

As is standard for the multiphase case, A_{pp}^{-1} is approximated using an AMG V-cycle in the first stage of CPR. The second stage preconditioner is chosen such that $P_2^{-1} \approx A^{-1}$ with an incomplete LU factorization method (ILU).

In addition to the two-stage preconditioner, recall that decoupling operators are often used to reduce the coupling between the pressure equation and the saturation variables in multiphase case. In the single-phase case, an approximation of the pressure equation $A_{pp}\delta p + A_{pT}\delta T = b_T$ is performed in the first stage of CPR where the temperature coupling A_{pT} is ignored. The single-phase equivalents of the Quasi-IMPES (QI) and True-IMPES (TI) decoupling operators have decoupling coefficients $M_{QI} = \text{diag}(A_{pT})\text{diag}(A_{TT})^{-1}$, $M_{TI} = \text{colsum}(A_{pT})\text{colsum}(A_{TT})^{-1}$, respectively.

By performing this decoupling operation on the system (5.6), the first stage of CPR now consists in solving a subsystem for the approximate Schur complement $S_p = A_{pp} - MA_{Tp}$ instead of the original pressure block. However, the properties of the resulting S_p need to be amenable to the application of AMG. As discussed in Section 3.4, this is not guaranteed, especially for QI. For the test cases detailed in Section 5.2, we observe little difference in the performance of CPR with decoupling operators (results not shown here). Therefore, we will only show results for CPR without a decoupling operator.

5.1.2 Block factorization preconditioner

Consider the following decomposition of the Jacobian

$$A = \begin{bmatrix} I & 0 \\ A_{Tp}A_{pp}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix} \begin{bmatrix} I & A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix}, \tag{5.13}$$

where $S_T = A_{TT} - A_{Tp}A_{pp}^{-1}A_{pT}$ is the Schur complement. The inverse of the Jacobian is given by

$$A^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I \end{bmatrix}. \tag{5.14}$$

Even if A is sparse, the Schur complement S_T is generally dense. A common preconditioning technique is to use the blocks of the factorization (5.13) combined with a sparse approximation of the Schur complement [32]. The notion of Schur complement approximation for general saddle-point problems is introduced in [68] and extended to general nonsymmetric matrices in [48]. Schur complement approximations appeared earlier for specific saddle-point problems, for example Stokes problems of incompressible fluid dynamics in [92]. Preconditioners of the following forms are considered in [48]:

$$P_L = \begin{bmatrix} I & 0 \\ A_{Tp}A_{pp}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix}, \text{ so that } P_L^{-1} = \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I \end{bmatrix} \tag{5.15}$$

$$P_U = \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix} \begin{bmatrix} I & A_{pp}^{-1} A_{pT} \\ 0 & I \end{bmatrix}, \text{ so that } P_U^{-1} = \begin{bmatrix} I & -A_{pp}^{-1} A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix}, \quad (5.16)$$

$$P_D = \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix}, \text{ so that } P_D^{-1} = \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix}, \quad (5.17)$$

where S_T is the exact Schur complement. The preconditioned matrices $P_L^{-1}A$, AP_L^{-1}, $P_U^{-1}A$, and AP_U^{-1} have a single eigenvalue $\lambda = 1$ since they are triangular matrices with a unit diagonal. These matrices also have a minimal polynomial of degree 2. Therefore, Krylov subspace iterative methods with an optimality or Galerkin property (such as GMRES) will terminate in at most 2 iterations [86]. In the case of saddle-point problems, $P_D^{-1}A$ and AP_D^{-1} have a minimal polynomial of degree 4, but this does not apply in our case. See [8] for a study where the Schur complement is not exact. In essence, a good Schur complement approximation leads to fewer iterations.

Since we want to focus on the quality of the Schur complement approximation, we will only show results for the full factorization preconditioner (5.14). While P_L, P_U, P_D usually result in more GMRES iterations than the full factorization, they exhibit similar scaling properties. It remains to be seen if one version is generally more efficient in terms of solution time.

Given an appropriate Schur complement approximation $\tilde{S}_T$, applying the block preconditioner (5.14) can be done as follows:

1. Solve the pressure subsystem: $A_{pp}x_p = b_p$;
2. Compute the new energy equation residual: $\tilde{b}_T = b_T - A_{Tp}x_p$;
3. Solve the Schur complement subsystem: $\tilde{S}_T\delta_T = \tilde{b}_T$;
4. Compute the new mass equation residual: $\tilde{b}_p = x_p - A_{pT}\delta_T$;
5. Solve the pressure subsystem: $A_{pp}\delta_p = \tilde{b}_p$.

In our case, A_{pp}^{-1} and $\tilde{S}_T^{-1}$ are both approximated using an AMG V-cycle.

5.1.3 Schur complement approximation

Common sparse approximations for the Schur complement are $\tilde{S}_{A_{TT}} = A_{TT}$ and $\tilde{S}_{\text{diag}} = A_{TT} - A_{Tp}\text{diag}\left(A_{pp}\right)^{-1}A_{pT}$. Here we present a Schur complement approximation which performs significantly better than such simple approximations.

For the derivation of our Schur complement approximation, we consider the linearized problem before discretization. This approach results in an approximation which holds as we refine the mesh. See [62] for a theoretical framework in using the infinite-dimensional setting to find mesh-independent preconditioners for self-adjoint problems.

We begin with a simple steady-state case and progressively reintroduce terms from the full problem.

5.1.3.1 Steady-state case

We first consider a steady-state single-phase thermal problem: find p, T such that

$$\nabla \cdot (\rho \mathbf{u}) = 0 \quad \text{in } \Omega, \tag{5.18}$$

$$\nabla \cdot (\rho c_v T \mathbf{u}) - \nabla \cdot (k_T \nabla T) = 0 \quad \text{in } \Omega, \tag{5.19}$$

where $\mathbf{u}$ is given by (2.2), and we have no-flux boundary conditions for the fluid and heat. Here we will consider the linearized system in a continuous setting. Applying a Newton method to (5.18)-(5.19), we obtain a block system of the form (5.6) where the blocks are

$$A_{pp} = \nabla \cdot (\rho \mathbf{u})_p, \qquad A_{pT} = \nabla \cdot (\rho \mathbf{u})_T, \tag{5.20}$$

$$A_{Tp} = \nabla \cdot (c_v T (\rho \mathbf{u})_p) = c_v \left[\nabla T \cdot (\rho \mathbf{u})_p + T \nabla \cdot (\rho \mathbf{u})_p \right], \tag{5.21}$$

$$\begin{aligned} A_{TT} &= c_v \left[\nabla \cdot (\rho \mathbf{u}) + \nabla \cdot (T (\rho \mathbf{u})_T) \right] - \nabla \cdot (k_T \nabla) \\ &= c_v \left[\nabla \cdot (\rho \mathbf{u}) + \nabla T \cdot (\rho \mathbf{u})_T + T \nabla \cdot (\rho \mathbf{u})_T \right] - \nabla \cdot (k_T \nabla), \end{aligned} \tag{5.22}$$

where we have used the product rule for the divergence operator in (5.21) and (5.22). Then the second term of the Schur complement (which corresponds in the continuous setting to the Poincaré-Steklov operator) becomes

$$\begin{aligned} A_{Tp} A_{pp}^{-1} A_{pT} &= c_v \left[\nabla T \cdot (\rho \mathbf{u})_p + T \nabla \cdot (\rho \mathbf{u})_p \right] \left(\nabla \cdot (\rho \mathbf{u})_p \right)^{-1} \nabla \cdot (\rho \mathbf{u})_T \\ &= c_v T \nabla \cdot (\rho \mathbf{u})_T + c_v \nabla T \cdot (\rho \mathbf{u})_p \left(\nabla \cdot (\rho \mathbf{u})_p \right)^{-1} \nabla \cdot (\rho \mathbf{u})_T. \end{aligned} \tag{5.23}$$

We notice that in $A_{TT} - A_{Tp} A_{pp}^{-1} A_{pT}$, the terms $c_v T \nabla \cdot (\rho \mathbf{u})_T$ cancel. We are left with

$$S_T = c_v \nabla \cdot (\rho \mathbf{u}) + c_v \nabla T \cdot (\rho \mathbf{u})_T - \nabla \cdot (k_T \nabla) - c_v \nabla T \cdot (\rho \mathbf{u})_p \left(\nabla \cdot (\rho \mathbf{u})_p \right)^{-1} \nabla \cdot (\rho \mathbf{u})_T. \tag{5.24}$$

One of the linearization terms has cancelled, and so we consider if it is possible that the last two terms also cancel. Consider the operator

$$(\rho \mathbf{u})_p \left(\nabla \cdot (\rho \mathbf{u})_p \right)^{-1} \nabla \cdot, \tag{5.25}$$

which appears in the last term of the Schur complement (5.24). This operator is close to the simpler operator

$$\nabla(\nabla \cdot \nabla)^{-1}\nabla \cdot =: s. \tag{5.26}$$

This holds if $(\rho\mathbf{u})_p$ is close to $-\nabla$, i.e. if ρ is close to being constant with respect to p. Indeed, when $(\rho)_p = 0$, we have

$$(\rho\mathbf{u})_p = \rho(\mathbf{u})_p = -\frac{\rho}{\mu}\mathbf{K}(\nabla - (\rho)_p\mathbf{g}) = -\frac{\rho}{\mu}\mathbf{K}\nabla. \tag{5.27}$$

While this approximation holds for liquid water and hydrocarbons, it may be less applicable in the case of gases. Extending this to multiphase flow is straightforward and discussed in Chapter 6.

Assuming that the operator s is applied to a sufficiently smooth vector field $\mathbf{F}$, we can use the Helmholtz decomposition to decompose this field into the sum of its curl-free and divergence-free part

$$\mathbf{F} = -\nabla\Phi + \nabla \times \mathbf{A}, \tag{5.28}$$

where Φ is a scalar potential and $\mathbf{A}$ a vector potential. Since s removes the divergence-free part of a field, applying s to $\mathbf{F}$, we obtain

$$s\mathbf{F} = -\nabla(\nabla \cdot \nabla)^{-1}\nabla \cdot \nabla\Phi = -\nabla\Phi, \tag{5.29}$$

assuming that Φ satisfies the same boundary conditions as the operator $(\nabla \cdot \nabla)^{-1}$. Hence s is a projection to the curl-free subspace, and it acts like the identity operator when applied to curl-free vector fields.

We assume that this also holds for $(\rho\mathbf{u})_p \left(\nabla \cdot (\rho\mathbf{u})_p\right)^{-1} \nabla \cdot$. In (5.24), we see that this operator is applied to $(\rho\mathbf{u})_T$. Assuming that $(\rho)_T = 0$ or that gravity is absent, this operator is of the form $\gamma\nabla p$, where γ is a scalar (or block-diagonal tensor) field. In order for $\gamma\nabla p$ to be a curl-free vector field, we need

$$\nabla \times (\gamma\nabla p) = \nabla\gamma \times \nabla p = 0, \tag{5.30}$$

i.e. we need $\nabla\gamma$ and ∇p to be parallel vectors. In the discretized case, our grid satisfies an orthogonality property as mentioned in Chapter 4, and the gradients of p and γ are approximated using a two-point flux approximation. In this case, the gradients are always orthogonal to the facets, and thus parallel. Accordingly, we replace the operator $(\rho\mathbf{u})_p \left(\nabla \cdot (\rho\mathbf{u})_p\right)^{-1} \nabla \cdot$ by the identity and obtain the following Schur complement approximation:

$$\tilde{S}_T = c_v\nabla \cdot (\rho\mathbf{u}) - \nabla \cdot (k_T\nabla). \tag{5.31}$$

This is a linear advection-diffusion operator acting on the temperature space. Note that it is the operator A_{TT} in (5.22) but without the linearization terms.

Similar heuristic arguments for replacing $\nabla \cdot (\nabla \cdot \nabla)^{-1} \nabla$ by the identity operator in the case of the Stokes problem can be found, for example, in [61], and for the Navier-Stokes equations, in [50].

5.1.3.2 Source terms

Similarly, we consider the steady-state case with the addition of source/sink terms. In this case, production wells satisfy $f_T^{\text{prod}} = c_v T f_{\text{prod}}$, while injection wells satisfy $f_T^{\text{inj}} = c_v T_{\text{inj}} f_{\text{inj}}$. Thus,

$$\begin{aligned} S_T = &\nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) - c_v f_{\text{prod}} + c_v \nabla T \cdot (\rho \mathbf{u})_T \\ &+ c_v T \nabla \cdot (\rho \mathbf{u})_T - c_v T (f_{\text{prod}})_T - c_v T_{\text{inj}} (f_{\text{inj}})_T \\ &- (c_v \nabla T \cdot (\rho \mathbf{u})_p + c_v T \nabla \cdot (\rho \mathbf{u})_p - c_v T (f_{\text{prod}})_p - c_v T_{\text{inj}} (f_{\text{inj}})_p) \\ &(\nabla \cdot (\rho \mathbf{u})_p - (f_{\text{prod}})_p - (f_{\text{inj}})_p)^{-1} \left[\nabla \cdot (\rho \mathbf{u})_T - (f_{\text{prod}})_T - (f_{\text{inj}})_T\right]. \end{aligned} \tag{5.32}$$

Since the injection term is weighted by T_{inj}, we cannot directly cancel the $c_v T$ terms as in (5.24). However, after a certain amount of injection, T tends to T_{inj} where the injection well is located. Furthermore, in the infinite-dimensional setting, this effect will be instantaneous since the well terms are defined using a delta function in (2.36). Using this argument, we get

$$\begin{aligned} S_T \approx &\nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) - c_v f_{\text{prod}} + c_v \nabla T \cdot (\rho \mathbf{u})_T \\ &- (c_v \nabla T \cdot (\rho \mathbf{u})_p) \\ &(\nabla \cdot (\rho \mathbf{u})_p - (f_{\text{prod}})_p - (f_{\text{inj}})_p)^{-1} \left[\nabla \cdot (\rho \mathbf{u})_T - (f_{\text{prod}})_T - (f_{\text{inj}})_T\right]. \end{aligned} \tag{5.33}$$

Further assuming that the mass source/sink terms are almost constant in p and T, i.e. $(\rho)_T$ and $(\rho)_p$ are small and the injection/production rates are independent of pressure and temperature (which is the case when operating at a target rate), we ignore the derivatives of the source/sink terms. Then, using the same argument as for the steady-state case, we obtain the Schur complement approximation

$$\tilde{S}_T = \nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) - c_v f_{\text{prod}} \tag{5.34}$$

In the case where the source terms are heaters, we have $f = 0$, and
$f_T = U(T_{\text{heater}} - T) D_{\text{heaters}}$, where D_{heaters} is the sum of delta functions for the location

of heaters. The Schur complement is given by

$$S_T = \nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) + U D_{\text{heaters}} + c_v \nabla T \cdot (\rho \mathbf{u})_T \\ - c_v \nabla T \cdot (\rho \mathbf{u})_p \left(\nabla \cdot (\rho \mathbf{u})_p\right)^{-1} \left[\nabla \cdot (\rho \mathbf{u})_T\right].$$

We see that heaters do not affect the right-hand side term. Using the same argument as above, we get the approximation

$$\tilde{S}_T = \nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) + U D_{\text{heaters}}. \tag{5.35}$$

Again, this operator consists of A_{TT} without its linearization terms.

5.1.3.3 Time-dependent case

We now generalize our analysis to the time-dependent problem. We first consider the case without source/sink terms. The blocks are given by

$$A_{pp} = \phi \frac{1}{\Delta t}(\rho)_p + \nabla \cdot (\rho \mathbf{u})_p, \tag{5.36}$$

$$A_{pT} = \phi \frac{1}{\Delta t}(\rho)_T + \nabla \cdot (\rho \mathbf{u})_T, \tag{5.37}$$

$$A_{Tp} = \phi \frac{1}{\Delta t}(\rho)_p c_v T + \nabla \cdot (c_v T (\rho \mathbf{u})_p), \tag{5.38}$$

$$A_{TT} = \phi \frac{c_v(\rho + (\rho)_T T)}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla \cdot (c_v \rho \mathbf{u}) \\ + \nabla \cdot (c_v T (\rho \mathbf{u})_T) - \nabla \cdot (k_T \nabla). \tag{5.39}$$

The second term of the Schur complement is given by

$$\begin{aligned} A_{Tp} A_{pp}^{-1} A_{pT} &= c_v \left[\phi \frac{1}{\Delta t}(\rho)_p c_v T + \nabla T \cdot (\rho \mathbf{u})_p + T \nabla \cdot (\rho \mathbf{u})_p\right] \\ &\quad \left(\phi \frac{1}{\Delta t}(\rho)_p + \nabla \cdot (\rho \mathbf{u})_p\right)^{-1} \left[\phi \frac{1}{\Delta t}(\rho)_T + \nabla \cdot (\rho \mathbf{u})_T\right] \\ &= c_v T \left[\nabla \cdot (\rho \mathbf{u})_T + \phi \frac{1}{\Delta t}(\rho)_T\right] + c_v \nabla T \cdot (\rho \mathbf{u})_p \\ &\quad \left(\phi \frac{1}{\Delta t}(\rho)_p + \nabla \cdot (\rho \mathbf{u})_p\right)^{-1} \left[\phi \frac{1}{\Delta t}(\rho)_T + \nabla \cdot (\rho \mathbf{u})_T\right], \end{aligned} \tag{5.40}$$

and thus the Schur complement is

$$S_T = \phi \frac{c_v \rho}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla \cdot (c_v \rho \mathbf{u}) - \nabla \cdot (k_T \nabla) + c_v \nabla T \cdot (\rho \mathbf{u})_T \\ - c_v \nabla T \cdot (\rho \mathbf{u})_p \left(\phi \frac{1}{\Delta t}(\rho)_p + \nabla \cdot (\rho \mathbf{u})_p\right)^{-1} \left[\phi \frac{1}{\Delta t}(\rho)_T + \nabla \cdot (\rho \mathbf{u})_T\right]. \tag{5.41}$$

To justify further simplification, we need to assume that either ρ is almost constant in p and T, or that the time-step is very large. We get the following Schur complement approximation:

$$\tilde{S}_T = \phi\frac{c_v\rho}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla\cdot(c_v\rho\mathbf{u}) - \nabla\cdot(k_T\nabla) + c_v\nabla T\cdot(\rho\mathbf{u})_T. \tag{5.42}$$

In the case where we have source/sink terms, the Schur complement is given by

$$\begin{aligned} S_T = {} & \phi\frac{c_v\rho}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla\cdot(c_v\rho\mathbf{u}) - \nabla\cdot(k_T\nabla) + c_v\nabla T\cdot(\rho\mathbf{u})_T \\ & + c_v T\nabla\cdot(\rho\mathbf{u})_T + UD_{\text{heaters}} - c_v f_{\text{prod}} - c_v T(f_{\text{prod}})_T - c_v T_{\text{inj}}(f_{\text{inj}})_T \\ & - c_v\left[\phi\frac{1}{\Delta t}(\rho)_p T + \nabla T\cdot(\rho\mathbf{u})_p + T\nabla\cdot(\rho\mathbf{u})_p - T(f_{\text{prod}})_p - T_{\text{inj}}(f_{\text{inj}})_p\right] \\ & \left(\phi\frac{1}{\Delta t}(\rho)_p + \nabla\cdot(\rho\mathbf{u})_p - (f_{\text{prod}})_p - (f_{\text{inj}})_p\right)^{-1} \\ & \left[\phi\frac{1}{\Delta t}(\rho)_T + \nabla\cdot(\rho\mathbf{u})_T - (f_{\text{prod}})_T - (f_{\text{inj}})_T\right]. \end{aligned} \tag{5.43}$$

Using the same argument for the injection temperature T_{inj} as in Section 5.1.3.2, we get

$$\begin{aligned} S_T \approx {} & \phi\frac{c_v\rho}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla\cdot(c_v\rho\mathbf{u}) - \nabla\cdot(k_T\nabla) + c_v\nabla T\cdot(\rho\mathbf{u})_T \\ & + UD_{\text{heaters}} - c_v f_{\text{prod}} - c_v\left[\nabla T\cdot(\rho\mathbf{u})_p\right] \\ & \left(\phi\frac{1}{\Delta t}(\rho)_p + \nabla\cdot(\rho\mathbf{u})_p - (f_{\text{prod}})_p - (f_{\text{inj}})_p\right)^{-1} \\ & \left[\phi\frac{1}{\Delta t}(\rho)_T + \nabla\cdot(\rho\mathbf{u})_T - (f_{\text{prod}})_T - (f_{\text{inj}})_T\right]. \end{aligned} \tag{5.44}$$

Then, again assuming that the mass source/sink terms are independent of p and T, we obtain the following Schur complement approximation:

$$\tilde{S}_T = \phi\frac{c_v\rho}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \nabla\cdot(c_v\rho\mathbf{u}) - \nabla\cdot(k_T\nabla) + UD_{\text{heaters}} - c_v f_{\text{prod}}. \tag{5.45}$$

Again, this operator consists of A_{TT} from (5.10) without its linearization terms. We can obtain the discretized version of this operator from (4.32) by removing the terms depending on the previous time-step, and evaluating the nonlinear terms at the previous Newton iteration. We get the following bilinear operator:

$$\begin{aligned} S_e(\delta T, r) \coloneqq {} & \int_\Omega \phi c_v\frac{\rho\delta T}{\Delta t}r\,\mathrm{d}x + \int_\Omega (1-\phi)\rho_r c_r\frac{\delta T}{\Delta t}r\,\mathrm{d}x \\ & + \int_{\Gamma_{\text{int}}}[r]\{\!\{\mathbf{K}\}\!\}\frac{(\rho)^{\text{up}}}{(\mu)^{\text{up}}}(\delta T)^{\text{up}}\left(\frac{[p]}{\|h^+-h^-\|} - \{\rho\}\mathbf{g}\cdot\mathbf{n}_e\right)\mathrm{d}S \\ & + \int_{\Gamma_{\text{int}}}[r]\{\!\{k_T\}\!\}\frac{[\delta T]}{\|h^+-h^-\|}\,\mathrm{d}S + \int_\Omega(-c_v f_{\text{prod}} + UD_{\text{heaters}})\delta T\,\mathrm{d}x. \end{aligned} \tag{5.46}$$

In summary, via the simplifying arguments above, we are able to get useful Schur complement approximations. Since these are done at the infinite-dimensional level, we expect them to perform well independently of the mesh size. Furthermore, the absence of linearization terms from the Schur complement approximation makes its matrix properties more predictable than A_{TT} with respect to the application of multigrid methods. Since it is essentially an advection-diffusion operator, AMG should perform well if the problem is not too advection-dominated [96].

5.2 Numerical results

In this section, we perform numerical experiments for our block preconditioner and CPR. These are implemented using the open source Finite Element software Firedrake [77]. The linear algebra backend is the PETSc library [9], allowing efficient and parallel computations. Our custom block preconditioner and CPR are implemented through Firedrake's Python interface. Recent work from [52] allows us to easily assemble our Schur complement approximation preconditioner by providing the bilinear operator (5.46). We modified the custom preconditioner class from [52] to allow the use of matrix formats other than `matfree`. For example, the default `aij` matrix format allows for faster computations for lower order methods such as the one described in Chapter 4. Our implementation is available on GitHub[1].

For the block preconditioner, we use our Schur complement approximation (5.46), unless stated otherwise. Both the pressure block A_{pp} and the approximate Schur complement are approximately inverted using a V-cycle of AMG. We use BoomerAMG [46] from the hypre library [36] with default parameters, i.e. a symmetric-SOR/Jacobi relaxation scheme (one sweep up, one sweep down), Falgout coarsening, classical Ruge-Stüben interpolation, and Gaussian Elimination as the coarse grid solver. This implementation of AMG is very efficient in parallel.

For the second stage of CPR, we use ILU(0) as provided by PETSc. In parallel, we use block Jacobi with ILU(0) for each block (the partition is assigned when Firedrake does the discretization).

The nonlinear solver is Newton's method with line search, and the linear solver is right-preconditioned GMRES [87], restarted after 30 iterations. The convergence tolerance of Newton's methods is 10^{-8} for the relative function norm and relative step size norm. The convergence tolerance for GMRES is 10^{-10} for the relative residual norm for the tests in Section 5.2.1, and 10^{-5} for the tests in Sections 5.2.3 and 5.2.4.

[1] `https://github.com/tlroy/thermalporous`

For all cases, we consider a heavy oil with density and viscosity as described in Section 2.5. The other physical parameters are shown in Table 5.1. These parameters are representative of those used in commercial reservoir simulators. The pressures are taken from the SPE10 test case [25].

Table 5.1: Physical parameters for test cases

Initial pressure	4.1369×10^7 Pa
Conductivity of oil	$0.15\ \mathrm{W\,m^{-1}\,K^{-1}}$
Conductivity of rock	$1.7295772056\ \mathrm{W\,m^{-1}\,K^{-1}}$
Specific heat of oil	$2093.4\ \mathrm{J\,K^{-1}\,kg^{-1}}$
Specific heat of rock	$920\ \mathrm{J\,K^{-1}\,kg^{-1}}$
Density of rock	$2650\ \mathrm{kg\,m^{-1}}$

For all cases, we evaluate the performance of the methods by comparing the number of linear iterations per nonlinear iteration. We note that, for our proof-of-concept implementation, the cost of applying the block preconditioner is around two times more computationally expensive (in serial) than CPR. The difference may not be as significant in an optimized implementation. Note that the cost of applying the block preconditioner is also lower when only using some of the factors as suggested in Section 5.1.2.

5.2.1 SPE10 test cases

The domain is a square with dimensions 365.76×365.76 meters, and the mesh is 60×120. For permeability and porosity fields, we use the benchmark problem SPE10 [25]. This problem has a highly heterogeneous permeability field. We consider a 60×120 slice in the x-y plane direction. The permeability, which is isotropic in the x-y plane, is illustrated in Figure 5.1. We do not include gravity for the 2D simulations.

For the well case (W), we have one production well and one injection well. These are located in the upper half of the domain in the regions of high permeability. For the injection and production rates, we use the Peaceman well model. The bottom-hole pressure for the injection well is fixed at 6.895×10^7 Pa, and 2.7579×10^7 Pa for the production well. The maximum rate is set to $q = 1.8 \times 10^{-3}\mathrm{m^3 s^{-1}}$, although this is only achieved for the high permeability cases. The initial temperature in the reservoir is 288.706 K and the injection temperature is 422.039 K. For the heater case (H), heater placement is the same as for the well case, and so are the initial and heating temperatures. For the well and heater case (W+H), we combine both wells

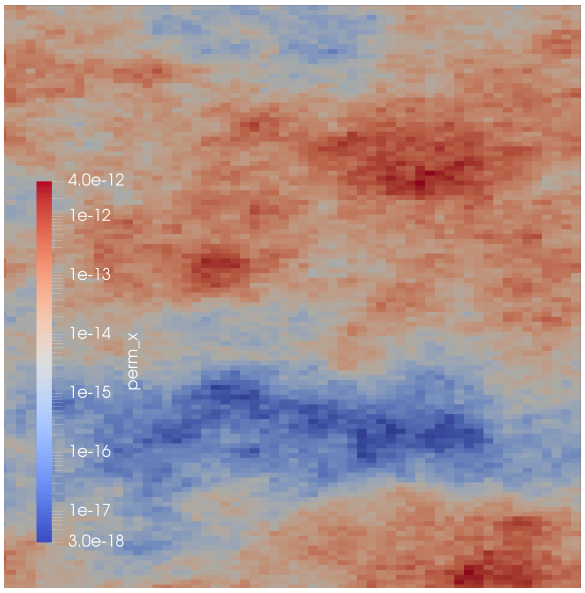


Figure 5.1: Log scale of the permeability of SPE10 test case (m^2).

and heaters. For the high permeability cases (h.p.), we increase the permeability by a factor of 1,000. A summary of the test case labels is given in Table 5.2. While the resulting permeability values are not representative of physical ones, they give a simple example of advection-dominated heat flow.

Table 5.2: Test case labels

W	Wells case
H	Heaters case
h.p.	High permeability

For each case, we simulate injection and production for 1000 days where the time-steps are chosen adaptively such that Newton's method converges in around 4 iterations. The average linear iterations per nonlinear iteration are shown in Table 5.3.

Table 5.3: SPE10 test cases. Average linear iterations per nonlinear iteration.

Method/Case	W	H	W+H	h.p. W	h.p. W+H
Block	5.88	5.42	6.60	14.5	14.0
CPR	6.67	6.27	6.69	11.4	11.0

We observe that for the first three cases in Table 5.3, the block preconditioner performs better than CPR in terms of the number of GMRES iterations, but that

CPR performs best for the high permeability cases. The heat flow for the first three cases is somewhat diffusion-dominated, especially when the oil is not yet heated. Note that heat diffusion is very slow relative to how large to grid blocks are. For the high permeability cases, advection easily dominates. This change in performance appears later in the simulation when the temperature has increased everywhere between the two wells. This indicates that CPR can still be a good choice if the temperature is simply transported by the fluid flow. However, we will see in the next sections that the block preconditioner is a more scalable method.

5.2.2 Numerical justification of the Schur complement approximation

We now perform a numerical comparison of the action of the inverses of the different Schur complement approximations. We use the cases given in Section 5.2.1. In Table 5.4, we compare the different Schur complement approximations by looking at the condition number of their inverse applied to the full Schur complement. While this condition number does not directly inform us about how well the preconditioner performs, it is a good indication of the quality of the approximations. For the cases, H and W stand for heaters and wells, respectively, and h.p. stands for high permeability (increased by a factor 1,000). We observe that $\tilde{S}_T$ is a good Schur complement approximation even for the high permeability cases where the other approximations struggle.

Table 5.4: Condition numbers (upper bounds) of the different matrices and Schur complement approximations for various cases

Matrix/Case	H	W	W+H	h.p. W	h.p. W+H
$\tilde{S}_{\text{diag}}^{-1}S$	1.061	20.75	3.323	8.703e7	2.191e7
$\tilde{S}_{\text{A}_{\text{TT}}}^{-1}S$	1.063	28.08	4.277	4117	2467
$\tilde{S}_T^{-1}S$	1.023	1.097	1.1717	5.969	5.939
A_{TT}	5.64e5	27.88	5.64e5	2324	2.143e5
S	5.479e5	2.862	5.717e5	20.60	4.976e5

In terms of the performance of the solver, $\tilde{S}_T$ always results in fewer GMRES iterations (results not shown here). For harder cases (for example high permeability), this difference is significant; the linear solver can even fail to converge before the prescribed maximum number of iterations. In the next section, we will see that the other Schur complement approximations struggle in anisotropic media.

5.2.3 Problem size scaling

We now investigate the performance of CPR and our block preconditioner as we refine a mesh. For two of the cases, we will also consider the Schur complement approximations $\tilde{S}_{A_{TT}}$ and $\tilde{S}_{\text{diag}}$. To this end, we test cases with homogeneous porosity and permeability fields. The domain is a square with dimensions 20×20 meters and uniform porosity $\phi = 0.2$. We test both isotropic and anisotropic permeability fields (but still homogeneous in each direction). We refine the mesh from a 20×20 grid to 320×320.

We begin with an isotropic permeability field of 3×10^{-13} m^2. For all cases, the injection/heating temperature is 422.039 K. For all cases except Case III, the initial temperature is 288.706 K. For each case, we take two time-steps and calculate the average number of linear iterations per nonlinear iteration. For Case I-IV, the time-step is 10 days, and for Case V, 12 hours.

For Case I, we have 6 heaters in the domain. In Table 5.5, we observe that the number of iterations increases by 9 times for CPR, while it increases by less than 50% for the block preconditioner.

Table 5.5: Case I: Heater case in isotropic medium. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
Block	2.57	3.23	2.86	3.44	3.71
CPR	3.4	5.38	9.09	16.3	30.7

For Case II, we have injection wells and 3 production wells. The wells operate at constant injection and production rates $q = 5 \times 10^{-8}$ m^3 s^{-1}. In Table 5.6, we observe that the number of iterations for CPR increases by 10 times while it only increases by less than 50 % for the block preconditioner with the Schur complement approximation $\tilde{S}_T$. We observe a similar increase in iterations for the block preconditioner with the Schur complement approximations $\tilde{S}_{A_{TT}}$ and $\tilde{S}_{\text{diag}}$.

For Case III, we also have 3 injection wells and 3 production wells. To allow higher rates and faster flow, we increase the initial temperature to 320K. The wells operate at injection and production rates $q = 10^{-6}$ m^3 s^{-1}. In Table 5.6, we observe that the number of iterations for CPR increases by more than 10 times while it only increases by around 50 % for the block preconditioner.

Table 5.6: Case II: Well case in isotropic medium. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
Block	2.43	2.43	2.86	3.28	3.71
CPR	3.71	5.71	9.86	19.4	37.4
Block ($\tilde{S}_{A_{TT}}$)	4.57	5	5.57	6.29	6.57
Block ($\tilde{S}_{\text{diag}}$)	4.14	4.43	5.29	5.86	6.43

Table 5.7: Case III: Higher injection well case in isotropic medium. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
Block	3.67	4.38	4.7	5.10	5.52
CPR	4.71	7.31	13.1	24.7	50.6

For case IV and V, we increase the permeability in the x-direction to 3×10^{-11} m^2. For Case IV, we have 6 heaters and observe the same trend as the previous case in Table 5.8.

Table 5.8: Case IV: Heater case in anisotropic medium. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
Block	2.31	2.67	3.25	3.67	3.86
CPR	3.11	4.56	8.56	15.8	30.4

For Case V, we have 3 injection wells and 3 production wells. The wells operate at constant injection and production rate $q = 1 \times 10^{-6}$ m^3 s^{-1}. In this case, the flow is much faster and thus the time-step size is reduced to half a day for the convergence of Newton's method. In Table 5.9, we observe that the number of iterations is doubled for the block preconditioner with $\tilde{S}_T$, increased by 4 times for CPR, and slightly less for the block preconditioner with $\tilde{S}_{A_{TT}}$. Additionally, the block preconditioner with $\tilde{S}_{\text{diag}}$ fails to converge within 200 GMRES iterations.

In summary, as we refine the mesh, the number of iterations has a very small increase for the block preconditioner, but a large increase for CPR. The heat diffusion is much more noticeable on fine meshes, which CPR does not treat appropriately. However, coarser meshes are common in commercial reservoir simulators.

Note that the success of the block preconditioner is also due to the linear scalability

Table 5.9: Case V: Well case in anisotropic medium. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
Block	2.38	3.27	4.52	4.68	5.36
CPR	2.86	3.6	4.76	7.0	12
Block ($\tilde{S}_{A_{TT}}$)	9	17.1	24.1	27.9	31.6
Block ($\tilde{S}_{\text{diag}}$)	> 200	> 200	> 200	> 200	> 200

of AMG for elliptic problems. By removing the need for ILU, we get a nearly mesh-independent preconditioner.

5.2.4 Parallel scaling

We now compare the performance of the two methods in parallel. We look at both weak and strong scaling.

5.2.4.1 Weak scaling

For weak scaling, we compare the parallel performance of the methods as we increase the number of processors and problem size. The domain is $50 \times 50 \times 50$ meters with an $N \times N \times N$ grid. Since this is a 3D case, we include gravity. The permeability is 3×10^{-13} m^2 and the porosity is 0.2. We seek to have around 100,000 degrees of freedom per processor. Thus, for the number of processors 1, 2, 4, 8, and 16, we have $N = 36, 46, 58, 73, 92$.

For the heating case, we have two sets of 21 heaters near the top and bottom of the domain. We take two time-steps of 100 days and illustrate the results in Table 5.10. We observe that the number of iterations increases by around 20 % for the block preconditioner and triples for CPR.

Table 5.10: Weak scaling: 3D Heating case. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
Block	7.5	7.9	8.25	8.75	9.29
CPR	15.75	22.3	29.9	38.5	45.6

For the well case, we have 21 injection wells near the top of the domain, and 21 production wells near the bottom. All wells operate at a constant injection/production

rate $q = 10^{-7}$ $\mathrm{m}^3\,\mathrm{s}^{-1}$. We take two time-steps of 10 days and illustrate the performance of the methods in Table 5.11. We observe that the number of iterations for the block preconditioner increases by around 40% while the number of iterations for CPR nearly triples.

Table 5.11: Weak scaling: 3D Well case. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
Block	4.29	4.43	4.71	5.25	5.89
CPR	6.57	10.0	12.3	16.3	18.4

5.2.4.2 Strong scaling

We use the same problem as the previous section on the finest mesh. We keep the problem size fixed while increasing the number of processors. For reservoir simulation, strong scaling is usually more relevant than weak scaling. Indeed, reservoir models often come with a (usually rather coarse) fixed grid. As observed in Section 5.2.3, CPR does not behave as well on a fine mesh. Thus, keeping the mesh size constant is a good way of isolating the parallel performance of the methods.

In Tables 5.12 and 5.13, we illustrate the strong scaling results for the heating and well cases, respectively. For the block preconditioner, we observe that the number of iteration is essentially independent of the number of processors used. This is thanks to the parallel capability of BoomerAMG. On the other hand, the number of iterations for CPR exhibit a small but progressive increase. This is because the second stage of CPR uses Block ILU, which becomes a weaker preconditioner as the number of blocks increases. Therefore, this trend will continue as the number of processors increases.

Table 5.12: Strong scaling: 3D Heating case. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
Block	8.75	8.57	8.57	9.43	9.29
CPR	38.0	43.3	44.0	44.7	45.6

5.2.5 Conclusion

In this chapter, we have implemented a fully implicit parallel non-isothermal porous media flow simulator including two preconditioning strategies, CPR and a block pre-

Table 5.13: Strong scaling: 3D Well case. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
Block	6.22	5.44	5.78	6.11	5.89
CPR	14.0	16.1	16.7	17.9	18.4

conditioner with our own Schur complement approximation. We have tested the performance of these methods as preconditioners for GMRES. Our Schur complement approximation performs better than the simple ones, especially in cases with heterogeneous or anisotropic permeability. While the block preconditioner performs well for diffusion-dominated cases, CPR is still the method of choice for advection-dominated (manufactured) cases, at least in serial. However, the block preconditioner scales nearly optimally with the problem size while CPR does not do well under mesh refinement. Additionally, the block preconditioner remains efficient in parallel, while the CPR iteration count increases gradually as we increase the number of processors.

The results in this chapter demonstrate that a preconditioning strategy which considers the diffusive effect of temperature is important for diffusion-dominated cases. In non-isothermal multiphase flow, the energy equation is treated in CPR like a hyperbolic equation. In the next chapter, we will present an extension of CPR where the diffusive effects of temperature are treated adequately.

Chapter 6

Preconditioning for Multiphase Flow

We investigate preconditioning strategies for the multiphase thermal model given by (2.18), (2.26), (2.19), (2.20), (2.27). We restate the problem here for convenience: find p, T, S_α for all phases $\alpha \neq \beta$ such that

$$\phi \frac{\partial (S_\alpha \rho_\alpha)}{\partial t} - \nabla \cdot \left(\rho_\alpha \frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha} (\nabla p - \rho_\alpha \mathbf{g}) \right) = f_\alpha \quad \text{in } \mathbb{R}_+ \times \Omega, \text{ for all } \alpha, \tag{6.1}$$

$$\frac{\partial}{\partial t} \left(\phi \sum_\alpha \rho_\alpha S_\alpha c_{v,\alpha} T + (1-\phi) \rho_r c_r T \right) - \nabla \cdot \sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha} (\nabla p - \rho_\alpha \mathbf{g}) - \nabla \cdot (k_T \nabla T) = f_T \quad \text{in } \mathbb{R}_+ \times \Omega, \tag{6.2}$$

$$- \left(\sum_\alpha \rho_\alpha c_{v,\alpha} T \mathbf{K} \frac{k_{r\alpha}}{\mu_\alpha} (\nabla p - \rho_\alpha \mathbf{g}) + k_T \nabla T \right) \cdot \mathbf{n} = g_N^T \text{ on } \Gamma_N^T, \; T = g_D^T \text{ on } \Gamma_D^T, \tag{6.3}$$

$$- \frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha} (\nabla p - \rho_\alpha \mathbf{g}) \cdot \mathbf{n} = g_{N,\alpha} \quad \text{on } \Gamma_N, \quad \text{for all } \alpha, \tag{6.4}$$

$$S_\alpha = g_{D,\alpha}, \text{ for all } \alpha, \quad p = g_D \quad \text{on } \Gamma_D. \tag{6.5}$$

We present an extension of CPR where a pressure-temperature subsystem is solved in the first stage. For this system, we extend the Schur complement approximation from the previous chapter. These results are also found in [83].

6.1 Preconditioning

The equations (4.46) can be written as a system of nonlinear equations for the real coefficients p_i, T_i, and $S_{o,i}$ of the DG(0) functions p^{n+1}, T^{n+1}, and S_o^{n+1}, respectively.

Let x be the vector of these coefficients and $\mathcal{F}$ the function such that $\mathcal{F}(x) = 0$ is equivalent to (4.46). By linearizing this equation with Newton's method, we must solve at each iteration

$$\frac{\partial \mathcal{F}}{\partial x} \mid_{x=x_k} (x_{k+1} - x_k) = -\mathcal{F}(x_k). \tag{6.6}$$

The resulting linearized systems can be written as a block system of the form

$$A\delta x = \begin{bmatrix} A_{pp} & A_{pT} & A_{ps} \\ A_{Tp} & A_{TT} & A_{Ts} \\ A_{sp} & A_{sT} & A_{ss} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta T \\ \delta s \end{bmatrix} = \begin{bmatrix} b_p \\ b_T \\ b_s \end{bmatrix} = b. \tag{6.7}$$

The blocks of the pressure-temperature submatrix are the discrete versions of

$$A_{pp} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_p + \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p - \sum_\alpha c_{v,\alpha} (f_\alpha)_p, \tag{6.8}$$

$$A_{pT} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_T + \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \sum_\alpha c_{v,\alpha} (f_\alpha)_T, \tag{6.9}$$

$$A_{Tp} \sim \phi \frac{1}{\Delta t} \sum_\alpha c_{v,\alpha} S_\alpha (\rho_\alpha)_p T + \sum_\alpha \nabla \cdot (c_{v,\alpha} T (\rho_\alpha \mathbf{u}_\alpha)_p) - (f_T)_p, \tag{6.10}$$

$$\begin{aligned} A_{TT} \sim \frac{\phi}{\Delta t} \sum_\alpha c_{v,\alpha} (\rho_\alpha + (\rho_\alpha)_T T) + (1-\phi) \frac{\rho_r c_r}{\Delta t} + \sum_\alpha \nabla \cdot (c_{v,\alpha} \rho_\alpha \mathbf{u}_\alpha) \\ + \sum_\alpha \nabla \cdot (c_{v,\alpha} T (\rho_\alpha \mathbf{u}_\alpha)_T) - \nabla \cdot (k_T \nabla) - (f_T)_T, \end{aligned} \tag{6.11}$$

where

$$(\rho_\alpha \mathbf{u}_\alpha)_p = -\frac{\mathbf{K} k_{r\alpha}}{\mu_\alpha} \left(\rho_\alpha (\nabla - (\rho_\alpha)_p \mathbf{g}) + (\rho_\alpha)_p (\nabla p - \rho_\alpha \mathbf{g}) \right), \tag{6.12}$$

and

$$(\rho_\alpha \mathbf{u}_\alpha)_T = -\mathbf{K} k_{r\alpha} \left[\left(\frac{\rho_\alpha}{\mu_\alpha} \right)_T (\nabla p - \rho_\alpha \mathbf{g}) - \frac{\rho_\alpha}{\mu_\alpha} (\rho_\alpha)_T \mathbf{g} \right]. \tag{6.13}$$

All coefficients in (6.8)-(6.13) are evaluated at the previous Newton iterate $(p_k, T_k, S_{\alpha,k})$, and $(.)_p$ and $(.)_T$ denote the partial derivatives with respect to p and T, respectively.

We seek to solve the resulting linearized system (6.7) using iterative methods [86] preconditioned using two-stage preconditioners.

Two-stage preconditioners. Let P_1 and P_2 be two preconditioners for the linear system $Ax = b$ for which we have the action of their (generally approximate) inverses P_1^{-1}, and P_2^{-1}. Recall from Section 3.3.3 that applying a multiplicative two-stage preconditioner can be done as follows:

1. Precondition using P_1: $x_1 = P_1^{-1}b$;
2. Compute the new residual: $b_1 = b - Ax_1$;
3. Precondition using P_2 and correct: $x = P_2^{-1}b_1 + x_1$.

The action of the two-stage preconditioner can be written as

$$P^{-1} = P_2^{-1}(I - AP_1^{-1}) + P_1^{-1}. \tag{6.14}$$

6.1.1 Constrained pressure residual (CPR)

Recall the industry-standard preconditioner CPR from Section 3.3.3. Let R_p be the pressure restriction operator such that $R_pAR_p^\top = A_{pp}$. Then the first stage of CPR is given by

$$P_1^{-1} = R_p^\top(R_pAR_p^\top)^{-1}R_p. \tag{6.15}$$

The second stage preconditioner P_2 is given by ILU on the full system. Also recall that decoupling operators can be used in conjunction with CPR, as described in Section 3.3.3.1. We will consider their use within the first stage of CPR, which for a decoupling operator G becomes

$$P_1^{-1} = R_p^\top(R_pGAR_p^\top)^{-1}R_pG. \tag{6.16}$$

In our case, we also perform a weighted sum of the mass equations in (4.48). Algebraically, this is equivalent to choosing the decoupling coefficients as $c_{v,\alpha}$ for the mass conservation equation of phase α and 0 for the energy conservation equation. Alternatively, it can be viewed like Full Row Sum (FRS) (3.39) where the 1's on the first row are replaced by $c_{v,\alpha}$ for the column associated with the equation for phase α and 0 for the energy equation.

For our test cases, using decoupling operators can reduce the number of GMRES iterations needed. True-IMPES (TI) (3.35) is observed to be superior to Quasi-IMPES (QI) (3.31), which sometimes increases the number of GMRES iterations.

6.1.2 Constrained pressure-temperature residual (CPTR)

We introduce a CPR-like two-stage preconditioner where a pressure-temperature subsystem is solved approximately in the first stage using an extension of the block preconditioner from Sections 5.1.2 and 5.1.3 or an unknown-based AMG method. We call this method Constrained Pressure-Temperature Residual (CPTR).

Let the pressure-temperature submatrix be denoted as

$$A_{00} = \begin{bmatrix} A_{pp} & A_{pT} \\ A_{Tp} & A_{TT} \end{bmatrix}, \quad \text{so that} \quad A = \begin{bmatrix} A_{00} & A_{0s} \\ A_{s0} & A_{ss} \end{bmatrix}. \tag{6.17}$$

For CPTR, the first stage preconditioner P_1 is given by

$$P_1^{-1} = \begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.18}$$

where A_{00}^{-1} is an approximation of the action of the inverse of A_{00}.

In our case, we consider two approaches to approximate A_{00}^{-1}: an approximation given by a Schur complement factorization detailed in Section 6.1.3, and an unknown-based AMG method discussed in Section 6.1.4.

Similarly to CPR, we use ILU for the second stage preconditioner P_2.

Let $b_0 = [\, b_p \;\; b_T \,]^\top$. Applying CPTR to the right-hand side $b = [\, b_0 \;\; b_s \,]^\top$ can be done as follows:

1. Solve the pressure-temperature subsystem: $A_{00}x_0 = b_0$;
2. Compute the new residual: $\tilde{b} = \begin{bmatrix} b_0 \\ b_s \end{bmatrix} - \begin{bmatrix} A_{00} \\ A_{0s} \end{bmatrix} x_0$;
3. Precondition and correct: $\begin{bmatrix} \delta_0 \\ \delta_s \end{bmatrix} = P_2^{-1}\tilde{b} + \begin{bmatrix} \delta_0 \\ 0 \end{bmatrix}$.

Thus

$$\delta = P_2^{-1}\left(I - (A - P_2)\begin{bmatrix} A_{00}^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right) b. \tag{6.19}$$

Similarly to CPR, let R_0 be the pressure-temperature restriction operator such that $R_0 A R_0^\top = A_{00}$. Then the first stage of CPTR is given by

$$P_1^{-1} = R_0^\top (R_0 A R_0^\top)^{-1} R_0. \tag{6.20}$$

6.1.2.1 Decoupling operators

Similarly to CPR, an approximation is made in the first stage of CPTR where the saturation coupling block A_{0s} is ignored. To reduce this decoupling error, we extend the idea of decoupling operators to the pressure-temperature system. In that case, the first stage of CPTR consists in solving the system with the Schur complement approximation $S_o = A_{00} - MA_{s0}$, for M a block diagonal matrix of appropriate dimensions. Decoupling operators for CPTR also have the form

$$G = \begin{bmatrix} I & -M \\ 0 & I \end{bmatrix}, \tag{6.21}$$

except that the first row includes both the pressure and energy equations, while the first column includes the pressure and temperature unknowns. The extensions of Quasi-IMPES (QI) and True-IMPES (TI) for CPTR have decoupling coefficients $M_{\text{QI}} = \text{diag}(A_{0s})\text{diag}(A_{ss})^{-1}$, and $M_{\text{TI}} = \text{colsum}(A_{0s})\text{colsum}(A_{ss})^{-1}$, respectively.

We will consider the use of decoupling operators within the first stage of CPTR, so that it becomes

$$P_1^{-1} = R_0^\top (R_0 G A R_0^\top)^{-1} R_0 G. \tag{6.22}$$

Since the Schur complement approximation described in Section 5.1.3.2 relies heavily on the PDE structure, it cannot be combined with decoupling operators in an obvious way. Alternatively, AMG methods for systems of PDEs such as the ones discussed in Section 3.3.2 are purely algebraic, and thus could be applied to the modified pressure-temperature matrix S_0. It is unclear, however, if the properties of S_0 are guaranteed to be amenable to the application of AMG.

6.1.3 Block preconditioner for the pressure-temperature subsystem

In this section, we present a choice for the pressure-temperature solver A_{00}^{-1}. The decomposition for the single-phase case (5.13) can also be done for the similar pressure-temperature submatrix:

$$A_{00} = \begin{bmatrix} I & 0 \\ A_{Tp}A_{pp}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{pp} & 0 \\ 0 & S_T \end{bmatrix} \begin{bmatrix} I & A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix}, \tag{6.23}$$

where $S_T = A_{TT} - A_{Tp}A_{pp}^{-1}A_{pT}$ is the Schur complement. Recall that the inverse of A_{00} is given by

$$A_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & S_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I \end{bmatrix}. \tag{6.24}$$

As is done for the single-phase case in Section 5.1.2, we seek an efficient Schur complement approximation which will give us the following preconditioner for A_{00}:

$$\tilde{A}_{00}^{-1} = \begin{bmatrix} I & -A_{pp}^{-1}A_{pT} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{pp}^{-1} & 0 \\ 0 & \tilde{S}_T^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{Tp}A_{pp}^{-1} & I. \end{bmatrix}. \tag{6.25}$$

Both A_{pp}^{-1} and $\tilde{S}_T^{-1}$ are approximated using an AMG V-cycle.

Applying the block preconditioner is done the same as in Section 5.1.2. The CPTR method with this pressure-temperature solver involves strictly more work than the first stage of the CPR method described in Section 3.3.3. The additional work is essentially an additional application of the solver A_{pp}^{-1}, as well as the construction and application of the solver $\tilde{S}_T^{-1}$. One could also use block triangular versions of the block preconditioner, i.e. ignoring the left or right factor in (6.25) as mentioned in Section 5.1.2.

6.1.3.1 Schur complement approximation

The two simple sparse Schur complement approximations $\tilde{S}_{A_{TT}} = A_{TT}$ and $\tilde{S}_{\text{diag}} = A_{TT} - A_{Tp}\text{diag}\left(A_{pp}\right)^{-1} A_{pT}$ were shown to perform poorly for the single phase case in Section 5.2. Here, we extend the Schur complement approximation presented in the previous chapter to the multiphase case. This Schur complement approximation requires the construction of the pressure equation (4.48).

We first consider the steady-state problem for pressure and temperature: find p, T such that

$$\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = 0 \quad \text{in } \Omega. \tag{6.26}$$

$$\sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha) - \nabla \cdot (\mathbf{k}_T \nabla T) = 0 \quad \text{in } \Omega, \tag{6.27}$$

where $\mathbf{u}_\alpha$ is given by Darcy's law (2.16), and we have homogeneous Neumann boundary conditions. Note that (6.26) is the pressure equation obtained from the weighting (4.48). Here, we consider the linearized pressure-temperature sub-system in a infinite-dimensional setting. Applying a Newton method to (6.26)-(6.27), we obtain a block matrix of the form (6.17) where the blocks are

$$A_{pp} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \,, \tag{6.28}$$

$$A_{pT} = \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \,, \tag{6.29}$$

$$A_{Tp} = \sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha)_p = \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p + \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \,, \tag{6.30}$$

$$\begin{aligned} A_{TT} =& \sum_\alpha \nabla \cdot (\rho_\alpha c_{v,\alpha} T \mathbf{u}_\alpha)_T - \nabla \cdot (\mathbf{k}_T \nabla) \\ =& \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \\ &+ \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T - \nabla \cdot (\mathbf{k}_T \nabla), \end{aligned} \tag{6.31}$$

where the product rule was used for the divergence operator in (6.30) and (6.31). Then, the second term of the Schur complement (which corresponds to the Poincaré-Steklov operator in the infinite-dimensional setting) becomes

$$\begin{aligned} A_{Tp} A_{pp}^{-1} A_{pT} =& \left(\sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p + \sum_\alpha c_{v,\alpha} T \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right) \\ & \left(\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \\ =& T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \\ & \left(\sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \\ =& T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T + \nabla T \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \\ & \left(\nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_T \,. \end{aligned} \tag{6.32}$$

Taking the difference between (6.31) and (6.32), we obtain the Schur complement, $A_{TT} - A_{Tp} A_{pp}^{-1} A_{pT}$. We notice that the term $T \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha)_T$ appears in both (6.31) and (6.32) and thus cancels. This is due to the construction of the pressure equation (4.48). We are left with

$$\begin{aligned} S_T =& \sum_\alpha c_{v,\alpha} \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) - \nabla \cdot (\mathbf{k}_T \nabla) + \sum_\alpha c_{v,\alpha} \nabla T \cdot (\rho_\alpha \mathbf{u}_\alpha)_T \\ & - \nabla T \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \left(\nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_p \right)^{-1} \nabla \cdot \sum_\alpha c_{v,\alpha} (\rho_\alpha \mathbf{u}_\alpha)_T \,. \end{aligned} \tag{6.33}$$

Since one of the terms coming from the linearization has canceled, we investigate if it is possible that the last two terms in (6.33) can cancel under reasonable assumptions. Consider the following operator taken from the last term of (6.33):

$$\sum_\alpha c_{v,\alpha} \left(\rho_\alpha \mathbf{u}_\alpha\right)_p \left(\nabla \cdot \sum_\alpha c_{v,\alpha} \left(\rho_\alpha \mathbf{u}_\alpha\right)_p\right)^{-1} \nabla \cdot . \tag{6.34}$$

We now make the same approximations as in Section 5.1.3.1, which includes that ρ_α is constant with respect to p, and that we use a two-point flux approximation for the facet integrals. This allows us to replace the operator (6.34) by the identity, giving us the following simplified and more practical Schur complement approximation:

$$\tilde{S}_T = \sum_\alpha c_{v,\alpha} \nabla \cdot \left(\rho_\alpha \mathbf{u}_\alpha\right)_T - \nabla \cdot (\mathbf{k}_T \nabla). \tag{6.35}$$

While the assumption that ρ_α is close to being a constant with respect to p is appropriate for liquid oil and water, it may be less applicable in the case of gases. However, early investigations in Section 6.2.3.5 will indicate that increasing the compressibility of the fluid does not reduce the effectiveness of the preconditioner.

The extension to the time-dependent case and the addition of source/sink terms does not differ from the single-phase case in Sections 5.1.3.2 and 5.1.3.3. In this case, the source/sink terms related to production wells satisfy $f_{T,\text{prod}} = \sum_\alpha c_{v,\alpha} T f_{\alpha,\text{prod}}$, while those for injection wells satisfy $f_{T,\text{inj}} = \sum_\alpha c_{v,\alpha} T_{\text{inj}} f_{\alpha,\text{inj}}$. Also, the source term for heaters is given by $f_T = U(T_{\text{heater}} - T) D_{\text{heaters}}$, where D_{heaters} is the sum of delta functions for the locations of heaters. The Schur complement approximation is given by

$$\begin{aligned}\tilde{S}_T = \phi \sum_\alpha \frac{c_{v,\alpha}\rho_\alpha}{\Delta t} + (1-\phi)\frac{\rho_r c_r}{\Delta t} + \sum_\alpha c_{v,\alpha} \nabla \cdot \left(\rho_\alpha \mathbf{u}_\alpha\right)_T - \nabla \cdot (k_T \nabla) \\ + U D_{\text{heaters}} - \sum_\alpha c_{v,\alpha} f_{\alpha,\text{prod}}.\end{aligned} \tag{6.36}$$

The discretized version of this operator can be obtained from (4.42) by removing the terms that only depend on the previous time-step, and by evaluating the coefficients with the values of the previous Newton iterate. We get the following bilinear operator:

$$\begin{aligned}S_e(\delta T, r) \coloneqq & \sum_\alpha \int_\Omega \phi c_{v,\alpha} \frac{S_\alpha \rho_\alpha \delta T}{\Delta t} r \,\mathrm{d}x + \int_\Omega (1-\phi)\rho_r c_r \frac{\delta T}{\Delta t} r \,\mathrm{d}x \\ & + \sum_\alpha \int_{\Gamma_{\text{int}}} [r] \{\!\{\mathbf{K}\}\!\} (k_{r\alpha})^{\text{up}} c_{v,\alpha} \frac{(\rho_\alpha)^{\text{up}}}{(\mu_\alpha)^{\text{up}}} (\delta T)^{\text{up}} \left(\frac{[p]}{\|[h]\|} - \{\rho_\alpha\} \mathbf{g} \cdot \mathbf{n}_e\right) \mathrm{d}S \\ & + \int_{\Gamma_{\text{int}}} [r] \{\!\{k_T\}\!\} \frac{[\delta T]}{\|[h]\|} \,\mathrm{d}S + \int_\Omega \left(-\sum_\alpha c_{v,\alpha} f_{\alpha,\text{prod}} + U D_{\text{heaters}}\right) \delta T \,\mathrm{d}x.\end{aligned} \tag{6.37}$$

6.1.4 AMG for the pressure-temperature subsystem

Recall from Section 3.3.2 that there exist versions of AMG for systems of PDEs. While applying such methods to the full system is problematic since the equations are hyperbolic with respect to the saturations, we can instead apply an AMG method for systems to the restricted pressure-temperature subsystem. Indeed, the equations are elliptic with respect to pressure, and the energy equation is parabolic with respect to temperature. This makes the pressure-temperature subsystem a good candidate for AMG. We will consider an unknown-based AMG method for the pressure-temperature subsystem of the first stage of CPTR. In Section 6.2.3.5, we will also compare this to applying AMG to the pressure and temperature blocks independently.

6.1.5 Discussion on multi-stage preconditioners

The first stage of CPR is given by the approximate solution of $A_{pp}\widetilde{\delta p} = b_p$, which is an approximation of the pressure equation

$$A_{pp}\delta p + A_{pT}\delta T + A_{ps}\delta s = b_p. \tag{6.38}$$

If the coupling blocks A_{pT} and A_{ps} are significant, the approximate pressure $\widetilde{\delta p}$ might not be representative. This could be countered by reducing the coupling between the pressure equation and the non-pressure variables. Similarly, the first stage of CPTR is also currently vulnerable to a strong coupling between both the pressure and energy equations and saturation variables since it ignores both A_{ps} and A_{Ts}. We will consider the use of decoupling operators for both CPR and CPTR. However, for CPTR, this cannot be done when using our block preconditioner as the first-stage solver.

A simple way to reduce this coupling without the use of algebraic decoupling operators could be to change the order of the two-stage preconditioner. Then, the approximate solution of the fully coupled system (through ILU) acts as a decoupling operator. This idea has been used for example in [17] for the isothermal situation.

For our test cases, we do not observe a significant difference when using this variant of CPR. On the other hand, with the variant CPTR, we originally observed a significant decrease in the number of GMRES iterations in general, and an increase in the number of Newton iterations in some cases. Further investigation revealed that during GMRES, the residual of the energy equation was being reduced faster than those of the pressure equation and oil mass conservation equation. Then, since the energy equation has larger values when using SI units, the convergence threshold for the total residual was reached too early for the residual of the other equations to

have been reduced sufficiently. Even if Newton itself is scale-invariant, the Jacobian system is only being solved to a relative tolerance. This inadequate solution of the linearized system caused Newton to converge slower. An obvious remedy is to scale the equations such that they are of similar magnitude.

In our case, we multiply the pressure equation by a reference temperature, and the oil mass conservation equation by the product of the reference temperature and a reference specific heat coefficient. Thus, we modify F^* from (4.47) to

$$F^*(p, T, S_o; q, r, s) \coloneqq T_{\text{ref}} F_p(p, T, S_o; q) + F_e(p, T, S_o; r) + c_{\text{ref}} T_{\text{ref}} F_o(p, T, S_o; s). \quad (6.39)$$

This scaling has little effect on the number of iterations for the original CPR and CPTR. However, it makes variant CPTR perform essentially the same as the original CPTR. Thus, in the next section, we will only consider versions of CPR and CPTR with the original ordering of the stages. Nonetheless, we will use the scaling (6.39) by default.

As mentioned above, for variant CPTR, the residual for the energy equation is being reduced faster than the residuals of the mass equations. This indicates that decoupling the energy equation (in this case using ILU) improves the solution of the restricted pressure-temperature subsystem. However, the residual of the other equations is not reduced as fast for this method. We will investigate numerically in the next section if the use of decoupling operators can reduce the number of iterations for CPTR.

Finally, extending CPR and CPTR to more stages does not appear to have a significant effect on the number of iterations. For example, one could consider a three-stage preconditioner where ILU is used before and after a pressure (or pressure-temperature) restricted solution. Even in the cases where this results in fewer iterations, the additional step does not appear to be worth the additional cost. This is also true of the case where a pressure (-temperature) restricted solve is done before and after ILU.

6.2 Numerical results

We perform numerical experiments to evaluate the algorithmic performance of CPR and CPTR. The code used for these experiments can be found on GitHub[1]. This code is implemented using the open-source Finite Element software Firedrake [52,

[1] `https://github.com/tlroy/thermalporous`

77]. For solvers, Firedrake interfaces with the PETSc library [9], allowing for efficient and parallel computations. Our Schur complement approximation and the two-stage preconditioners are implemented through Firedrake's Python interface, including proof-of-concept implementations for the decoupling operators. Simple but less efficient versions of the two-stage preconditioners can also be implemented by simply providing PETSc options.

The first stage of CPTR is given by our Schur complement approximation (5.46) or by an unknown-based AMG. Unless stated otherwise, for both CPR and CPTR, an AMG V-cycle is used to approximate the inverse for the pressure block A_{pp}, the approximate Schur complement, and the temperature block A_{TT}. For both scalar and unknown-based AMG, BoomerAMG [46] from the hypre library [36] is used with default parameters, i.e. a symmetric-SOR/Jacobi relaxation scheme (one sweep up, one sweep down), Falgout coarsening, classical Ruge-Stüben interpolation, and Gaussian Elimination as the coarse grid solver. BoomerAMG is very efficient in parallel. For the second stage of CPR and CPTR, ILU(k) is used as provided in PETSc. Unless stated that we use ILU(1), ILU(0) is our default. The parallel version of ILU is block Jacobi where ILU(k) is used for each block (the blocks are determined when Firedrake assigns different parts of the domain to different processors).

The nonlinear solver is Newton's method with line search, and the linear solver is right-preconditioned GMRES [87]. For Newton's method, the convergence tolerance for both the relative function norm and relative step size norm is set to 10^{-8}. For GMRES, the convergence tolerance for the relative residual norm is set to 10^{-8}.

For all cases, we consider oil and water with densities and viscosities as described in Section 2.5. The other physical parameters are shown in Table 6.1. These parameters are representative of those used in commercial reservoir simulators. We also only consider homogeneous Neumann boundary conditions.

Table 6.1: Physical parameters for test cases

Initial pressure	4.1369×10^7 Pa
Conductivity of oil	0.15 $\mathrm{W\,m^{-1}\,K^{-1}}$
Conductivity of water	0.6005638 $\mathrm{W\,m^{-1}\,K^{-1}}$
Conductivity of rock	1.7295772056 $\mathrm{W\,m^{-1}\,K^{-1}}$
Specific heat of oil	2093.4 $\mathrm{J\,K^{-1}\,kg^{-1}}$
Specific heat of water	4181.3 $\mathrm{J\,K^{-1}\,kg^{-1}}$
Specific heat of rock	920 $\mathrm{J\,K^{-1}\,kg^{-1}}$
Density of rock	2650 $\mathrm{kg\,m^{-1}}$

For the scaling parameters in (6.39), we choose $T_{\text{ref}} = T_{\text{prod}}$ and $c_{\text{ref}} = S_o c_{v,o} + (1 - S_o)c_{v,w}$, where S_o is the initial oil saturation in the reservoir, which is uniform in all the test cases here. For the preconditioners we consider, the number of iterations is not very sensitive to the choice of scaling parameters.

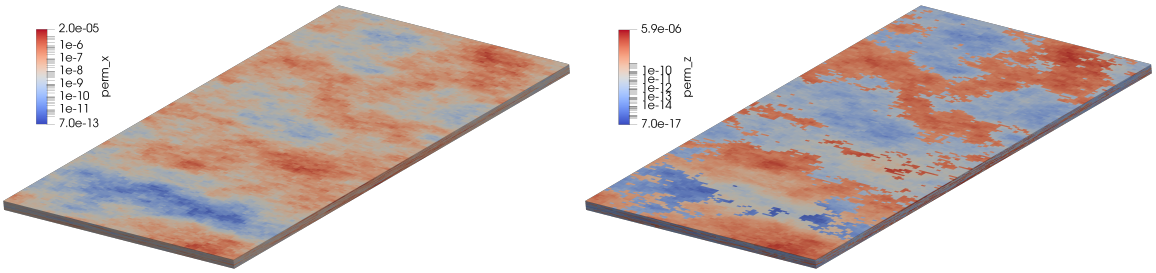
For all cases, the algorithmic performance of the methods is evaluated by comparing the number of linear iterations per nonlinear iteration.

Computational cost We note for the first stage of CPTR, applying the full block preconditioner from Section 6.1.3 should be slightly more than three times the cost of applying the first stage of CPR, while the unknown-based AMG should be around two times the cost. The relative cost of the second stage will depend on things such as the size and dimension of the problem, and the number of processors used. To provide one example, we look at the CPU timings for the serial case given in Table 6.12 (a problem with 52,728 degrees of freedom). Applying the first stage of CPR takes an average of 0.0088s, while the first stage of CPTR with the block preconditioner takes an average of 0.033s, and 0.017s for unknown-based AMG. Applying the second stage of CPR/CPTR with ILU(0) takes an average of 0.02s, and 0.034s with ILU(1). These timings were performed on a Lenovo ThinkCentre M920q with Intel(R) Core(TM) i5-8500T CPU @ 2.10GHz. Note that the implementation of the block preconditioner is not as optimized as the other first stage solvers. Additionally, the full factorization (5.45) used here may not be needed in all cases. Dropping one of the factors would make the cost of applying the block preconditioner slightly more than two times the cost of the first stage of CPR, or slightly more than unknown-based AMG.

6.2.1 SPE10 test case

We consider the benchmark problem SPE10 [25] and use its permeability and porosity fields. For the following tests, we consider the top 20 layers of SPE10, such that the domain has dimensions 365.76×670.56×12.192 meters, and the mesh is 60×220×20. This problem has a highly heterogeneous permeability field. The permeability is isotropic in the x-y plane, and anisotropic otherwise. The permeability fields are illustrated in Figure 6.1.

For our first test case, we position an injection well in the middle of the reservoir, and production wells in each corner. The wells are completed throughout the 20 layers. For the injection and production rates, we use the Peaceman well model. The production wells produce with a bottom-hole pressure of 2.7579×10^7 Pa. The injection well injects hot water with a maximum rate of $q = 1.8 \times 10^{-3}$ $\text{m}^3\,\text{s}^{-1}$. The



(a) Permeability in x and y directions. (b) Permeability in z direction

Figure 6.1: Log of permeability of the SPE10 test case (mm^2).

initial temperature in the reservoir is 288.706 K and the injection temperature is 373.15 K. For the heater case (H), heater placement is the same as for the well case, and so are the initial and heating temperatures. For the well and heater case (W+H), we combine both wells and heaters.

For each case, we simulate injection and production for 100 days where the time-steps are chosen adaptively such that Newton's method usually converges in less than 10 iterations. For each case, we look at the performance of different methods as we increase the number of processors from 1 to 16. We consider CPR and CPTR where A_{pp}^{-1} and $\tilde{S}_T^{-1}$ are approximated by either AMG V-cyle or direct LU factorization, denoted CPR(AMG) and CPTR(AMG), or CPR(LU) and CPTR(LU), respectively. We include the latter since we believe that there are cases where the AMG V-cycles in the first stage of CPR/CPTR are not as effective as usual. Since CPTR relies on three AMG V-cycles, it is likely more susceptible to any such weaknesses.

For the case with both wells and heaters, we take a total of 104 time-steps. The average linear iterations per nonlinear iteration are shown in Table 6.2. We observe that all methods have a small increase in iterations. At the beginning of the simulation, heating is the most significant effect. For the first time-steps, CPTR(AMG) takes fewer iterations than CPR(AMG), but the total number of iterations becomes the same at the 30th time-step (around 12 days). In contrast, CPTR(LU) has fewer iterations than CPR(LU). This indicates that single AMG V-cycles are not enough in this case to provide good pressure and temperature solutions in the first stage of CPTR(AMG).

For the case with only wells, we take a total of 101 time-steps. The average linear iterations per nonlinear iteration are shown in Table 6.3. We again observe that both CPR and CPTR have a small increase in iterations. For the few first time-steps, CPTR(AMG) takes fewer iterations than CPR(AMG), but the total number

Table 6.2: Strong scaling SPE10 3D wells and heaters case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR(AMG)	11.5	11.6	11.6	11.6	11.8
CPTR(AMG)	11.8	12.2	12.2	12.2	12.3
CPR(LU)	10.2	10.3	10.2	10.2	10.4
CPTR(LU)	10	9.91	9.91	10	10.1

of iterations becomes the same at the 10th time-step (around 4 minutes). Again, CPTR(LU) has fewer iterations than CPR(LU), which indicates that the additional AMG V-cycles of CPTR(AMG) are not helping in this case.

Table 6.3: Strong scaling SPE10 3D wells case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR(AMG)	11.7	11.9	11.8	11.8	12
CPTR(AMG)	12.1	12.4	12.4	12.4	12.6
CPR(LU)	10.4	10.5	10.4	10.3	10.4
CPTR(LU)	10.2	10	10	10.1	10.1

For the case with only heaters, we take a total of 45 time-steps, except for CPR(AMG) with 8 processors, where 46 time-steps are used. In that case, GMRES fails to converge within 200 iterations so we take a smaller time-step. A large jump in GMRES iterations for that same Newton step also appears for CPR(AMG) with 2 processors (177 iterations), CPR(AMG) with 16 processors (94 iterations), and CPTR(AMG) with 16 processors (168 iterations). Further investigation revealed that some values of the saturation variable S_o were greater than one during that Newton step. Since a negative water saturation (1-S_o) causes the A_{pp} block to become indefinite, the AMG coarse grid will not be representative and will ruin the performance of CPR(AMG) and CPTR(AMG). Using smaller time-steps or damping factors for Newton are standard strategies for avoiding negative saturations. As for the previous cases, we see that CPTR(LU) takes slightly fewer iterations than CPR(LU).

The SPE10 test case was not originally designed for thermal cases, but rather for testing upscaling techniques. Heat diffusion is not very significant due to the coarseness of the grid. The difficulty of the SPE10 test case rather lies in its highly heterogeneous permeability field. Instead of considering cases with such a coarse grid, in Section 6.2.3, we will consider finer grids and thus diffusion-dominated flows.

Table 6.4: Strong scaling SPE10 3D heaters case, $S_o = 0.9$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR(AMG)	8.55	9.79	8.8	8.8	9.28
CPTR(AMG)	8.5	8.66	8.75	8.75	9.66
CPR(LU)	8.31	8.37	8.37	8.39	8.36
CPTR(LU)	8.2	8.3	8.3	8.34	8.32

6.2.2 Numerical justification of the Schur complement approximation

We now compare the action of the inverses of the different Schur complement approximations. This is done for cases similar to the single-phase cases described in Section 5.2.1. The only difference is that we start with $S_o = 0.9$, and inject water with a temperature of 373.15 K. Recall that the domain is a 2D slice of SPE10 with a 60×120 grid. For the well case (W), there is one production well and one injection well. These are located in regions of high permeability. For the heater case (H), heater placement is the same as for the well case, and for the well and heater case (W+H), we combine both wells and heaters. For the high permeability cases (h.p.), permeability is increased by a factor of 1,000 to emulate advection-dominated heat flow. The test case labels are recalled in Table 6.5

Table 6.5: Test case labels

W	Wells case
H	Heaters case
h.p.	High permeability

For different Schur complement approximations, we look at the condition number of their inverse applied to the full Schur complement. While the condition number is not fully indicative of the performance of preconditioned GMRES, it informs us about the quality of the approximations. In Table 6.6, we observe that our custom approximation $\tilde{S}_T$ approximates the Schur complement well, even for the high permeability cases where the simpler approximations struggle.

In terms of the actual number of GMRES iterations, $\tilde{S}_T$ always performs the best for CPTR (results for the single-phase case shown in Section 5.2.3). This difference is even more significant for harder cases (for example high permeability). In some cases, GMRES does not converge before the prescribed maximum number of iterations

Table 6.6: Condition numbers (upper bounds) of the different matrices and Schur complement approximations for various cases ($S_o = 0.9$)

Matrix/Case	H	W	W+H	h.p. W	h.p. W+H
$\tilde{S}^{-1}_{\text{diag}}S$	1.069	6.31E+02	7.34E+04	3.04E+08	1.50E+09
$\tilde{S}^{-1}_{\text{A}_{\text{TT}}}S$	1.046	2.07E+01	2.44E+01	1.10E+01	1.44E+05
$\tilde{S}^{-1}_{T}S$	1.021	1.078	1.155	2.496	1.19E+02
A_{TT}	5.43E+05	2.18E+01	5.03E+03	1.99E+03	1.46E+05
S	5.28E+05	2.49E+00	5.30E+03	5.04E+02	1.59E+05

when using other approximations. As for the single-phase case, the other Schur complements exhibit worse performance for heterogeneous or anisotropic permeability fields.

6.2.3 Homogeneous test cases

We now compare the performance of several methods for problems with homogeneous permeability. We begin by a brief mesh refinement study for 2D cases. Then, we look at both weak and strong scaling for 3D cases. For weak scaling, the problem size is increased proportionally with the number of processors, while for strong scaling, the problem size remains fixed. For weak scaling, the problem size is increased via uniform mesh refinement.

For traditional reservoir simulation, strong scaling is often more relevant than weak scaling. Indeed, reservoir models are typically given with geological properties on a (usually rather coarse) fixed grid, for example, the SPE10 test case. Another good reason to study strong scaling is to isolate the effects of parallelization from the effects of mesh refinement observed for weak scaling. Nonetheless, the effects of mesh refinement are very relevant in the context of reservoir simulation, for example around wells or other features of interest.

6.2.3.1 Mesh refinement study

We perform a mesh refinement study for 2D test cases. The domain is 50×50 meters with an $N \times N$ grid. Starting at $N = 20$, we double N until we reach $N = 320$. Since this is a 2D case, we do not include gravity. For the isotropic cases, the permeability is 3×10^{-13} m^2 and the porosity is 0.2. For the anisotropic case, the permeability in the x-direction is 3×10^{-10} m^2 and in the y-direction, 3×10^{-13} m^2.

We consider three cases: one with heaters, and two with wells (isotropic and anisotropic). For each case, we have 6 source/sink terms with 3 located on either side of the square domain. In the heating case, these terms are all heaters, while for the well cases, one side has injection wells and the other, production wells. For the well cases, the production/injection rates are constant at $q = 3 \times 10^{-7}$ $\mathrm{m}^3\,\mathrm{s}^{-1}$, and water is injected at a temperature of 373.15 K. For the heater case, we have a heating temperature of 373.15 K. The initial oil saturation for both cases is set to $S_o = 0.9$. We then take two time-steps of 10 days.

For CPTR, we will compare the use of the following first stage solvers: block preconditioner (CPTR(Block)), unknown-wise AMG (CPTR(uAMG)), unknown-wise AMG with the True-IMPES decoupling operator (CPTR(uAMG-TI)). ILU(0) is used in the second stage. For CPR, we will compare the use of non-decoupled CPR with ILU(0) (CPR) and with ILU(1) (CPR-ILU(1)), as well as CPR with True-IMPES and ILU(0) (CPR(TI)).

For each case, we calculate the average number of linear iterations per nonlinear iteration and these are shown for the isotropic well case, the heater case, and the anisotropic well case in Tables 6.7, 6.8, and 6.9, respectively. In all cases, we observe that for smaller values of N, CPR and CPTR perform similarly. However, as we refine the mesh, the number of iterations for CPR increases significantly, while it does not increase much for CPTR, or even decreases in the heater case. In the well cases, this increase is least significant for CPTR(Block). In the heater case, CPTR(uAMG-TI) has the smallest number of iterations.

CPR treats temperature like a saturation, essentially assuming that heat is simply being transported by the fluid flow. When this is the case, ILU is sufficient. Since heat diffusion becomes more noticeable as the mesh is refined, the CPR strategy does not hold for finer meshes, and ILU is not enough. In contrast, the treatment of temperature for CPTR does not depend on ILU, but rather AMG. Hence, CPTR tackles heat diffusion appropriately on fine meshes.

6.2.3.2 Weak scaling

For weak scaling, the performance of the methods is evaluated as we increase the number of processors and problem size simultaneously. The domain is a $50 \times 50 \times 50$ meters box, discretized with an $N \times N \times N$ grid. The permeability is constant at $3 \times 10^{-13}\mathrm{m}^2$ and the porosity is 0.2. For the number of processors 1, 2, 4, 8, and 16, we have $N = 26$, 33, 41, 52, 65, respectively. This results in around $50,000$ degrees of freedom per processor.

Table 6.7: 2D Isotropic Well case. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
CPR	5.29	6.29	9.14	15.5	31.5
CPR(TI)	4.86	5.43	8.29	14.7	30.7
CPR-ILU(1)	5.29	5.43	6.43	10	20
CPTR(Block)	5.29	5.29	5.43	5.64	6.53
CPTR(uAMG)	5.29	5.29	6.29	6.82	8.24
CPTR(uAMG-TI)	4.29	5.14	5.29	5.82	7.35

Table 6.8: 2D Heater case. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
CPR	5.33	5.5	8.5	14	25.8
CPR(TI)	4.5	5.17	7.83	13.6	25.2
CPR-ILU(1)	6.71	6.86	7.67	12.2	21.6
CPTR(Block)	4.83	5.17	5.17	4.6	5
CPTR(uAMG)	4.67	5	5	5	5
CPTR(uAMG-TI)	4.33	4.33	4.33	4.2	4.4

For the cases here, the different versions of CPR and CPTR tested in Section 6.2.3.1 perform similarly. We therefore only show results for non-decoupled CPR and CPTR (with the block preconditioner).

We consider three cases: one with heaters, and two with wells. For each case, we have 21 source/sink terms both near the top and bottom of the domain. In the heating case, these terms are all heaters, while for the well cases, the top terms are injection wells and the bottom terms, production wells. For the well cases, the production/injection rates are constant at $q = 10^{-7}$ $\mathrm{m^3\,s^{-1}}$. Water is injected at a temperature of 373.15 K.

The initial saturation for the heating case is set to $S_o = 0.5$, and for the well

Table 6.9: 2D Anisotropic Well case. Average linear iterations per nonlinear iteration.

Method/N	20	40	80	160	320
CPR	5	6	9	17	32.3
CPR(TI)	4.29	5.71	8.25	16.1	31.8
CPR-ILU(1)	4.71	5.43	6.5	11.1	20.4
CPTR(Block)	4.43	5.14	5.75	6.92	7.17
CPTR(uAMG)	4.86	5.43	6.25	8.08	10.1
CPTR(uAMG-TI)	4.14	5	5.5	7.33	9.44

cases to $S_o = 0.99$ and $S_o = 1$. For each case, we take 5 time-steps and look at the average number of linear iterations per nonlinear iteration. The size of the time-steps is $\Delta t = 10$ days for the heating case and the well case with $S_o = 0.99$, and $\Delta t = 4$ days for the well case with $S_o = 1$.

Table 6.10: Weak scaling: 3D Heating case, $S_o = 0.5$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	6.06	7.47	8.88	10.1	11.8
CPTR	5.47	5.47	5.88	5.88	6
CPR-ILU(1)	5.53	7.41	8.71	9.65	10.9

Table 6.11: Weak scaling: 3D Well case, $S_o = 0.99$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	6	7.78	8.31	9.54	12.4
CPTR	5.58	5.67	5.85	6.08	6.55
CPR-ILU(1)	5.58	7.33	8	9.08	11.8

The results for the heating case are shown in Table 6.10. We observe that as we increase the number of processors from 1 to 16, the number of iterations for both versions of CPR nearly doubles, while the number of iterations for CPTR increases by less than 10%. The results are similar for the well case with $S_o = 0.99$, shown in Table 6.11. The number of iterations for both versions of CPR more than doubles, and the number of iterations for CPTR increases by less than 20%. Note that for both heating and well cases, different starting constant saturations provide similar scaling results, except when S_o is greater than 0.999 for the well case.

Table 6.12: Weak scaling: 3D Well case, $S_o = 1$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	14.1	16	17.3	20.6	23.5
CPTR	14.1	15.9	17.3	20.5	23.4
CPR-ILU(1)	11.7	14	15.3	17.7	19.3

As shown in Table 6.12, the results are significantly different for the well case with $S_o = 1$. Indeed, the number of iterations for all methods increases by the same

factor of 2/3. We observe that CPTR performs essentially the same as CPR (but at an additional cost). This could indicate that the temperature solution given by the first stage of CPTR is either wrong or trivial. Additionally, CPTR scales better in Tables 6.10 and 6.11, which suggests again that there is a large decoupling error in the first stage of CPTR. As discussed in Section 6.1.5, we suspect this is due to a strong coupling between the temperature and saturation. Indeed, the energy equation solved in the first stage of CPTR uses S_o from the previous iteration, which may be a poor approximation when injecting hot water in a reservoir saturated with cold oil.

Furthermore, the well case with $S_o = 1$ is very similar to some single-phase test cases in Section 5.2.4 except that oil was injected instead of water. In Chapter 5, the block preconditioner scales significantly better than CPR in both weak and strong scaling. This is a clear indication that the issue with CPTR is related to the multiphase situation, possibly due the coupling of the saturation variable. However, using True-IMPES with CPTR does not change its poor scaling behaviour.

6.2.3.3 Strong scaling

We use the same problem as the previous section on the finest mesh ($N = 65$). We keep the problem size fixed while increasing the number of processors.

Table 6.13: Strong scaling: 3D Heating case, $S_o = 0.5$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	9.23	10.8	11.2	11.4	11.9
CPTR	5.77	5.85	5.92	6.08	6.38
CPR-ILU(1)	6.85	9.69	10.2	10.8	10.9

Table 6.14: Strong scaling: 3D Well case, $S_o = 0.99$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	9.45	11.4	12	12.2	12.4
CPTR	6.15	6.3	6.35	6.3	6.55
CPR-ILU(1)	6.8	10.4	11	11.2	11.8

In Table 6.13, we observe for the heating case that as we increase the number of processors from 1 to 16, the number of iterations increases by around 30% for CPR, 60% for CPR-ILU(1) and 10% for CPTR. The bulk of the increase for CPR-ILU(1)

occurs when going from 1 to 2 processors, around 40%. This showcases how CPR-ILU(1) is much more effective in serial. We observe similar results for the well case with $S_o = 0.99$ in Table 6.14. Increasing the number of processors from 1 to 16, the number of iterations increases by around 31% for CPR, 73% for CPR-ILU(1) and 7% for CPTR. Again, there is a large increase in the number of iterations for CPR-ILU(1) when going from 1 to 2 processors.

Table 6.15: Strong scaling: 3D Well case, $S_o = 1$. Average linear iterations per nonlinear iteration.

Method/Num. proc.	1	2	4	8	16
CPR	22	22.8	22.4	22.6	23.5
CPTR	22.1	22.8	22.3	22.5	23.4
CPR-ILU(1)	16.6	18	18.1	18.8	19.3

In Table 6.15, we show the results for the well case for $S_o = 1$. We observe a similar increase in the number of iterations for CPR and CPTR of 7% and 6%, respectively. For CPR-ILU(1), the increase in iterations is of 17%, and half of this increase happens when going from 1 to 2 processors.

Both CPR and CPTR use Block ILU, which becomes weaker as the number of processors increases. In contrast, AMG should scale much better. In cases other than $S_o = 1$, CPTR scales much better than CPR. For those cases, the first stage of CPTR gives an accurate temperature update, while CPR relies on ILU for its temperature update. In the case with $S_o = 1$, the challenge for solver is phase displacement rather than thermal effects.

6.2.3.4 Computational time

We compare the computational time of the different methods for cases similar to those studied above, but on finer meshes. These computations were performed on arcus-b, the largest compute cluster of Oxford's Advanced Research Computing services. Our computations are done with up to 16 cores per node. Our computations with 32 and 64 cores are done on 2 and 4 nodes, respectively. We will compare CPR with CPTR using both the block preconditioner and unknown-based AMG (uAMG) in its first stage. ILU(0) is used in the second stage of each method.

Consider the weak scaling cases in Section 6.2.3.2 for which the iteration counts are shown in Tables 6.10 and 6.11. We now consider $N \times N \times N$ meshes which result in around 100,000 degrees of freedom per processor so that we have $N = 32$,

41, 52, 65, 82, 103, and 129, for numbers of processors 1, 2, 4, 8, 16, 32, and 64, respectively. The largest case has 6,440,067 degrees of freedom. We compare the total computational time taken for the five time-steps, the total number of linear iterations, and the average time per linear iteration. These are shown in Tables 6.16 to 6.18 for the heating case, and in Tables 6.19 to 6.21for the well case.

Table 6.16: Weak scaling: 3D Heating case, $S_o = 0.5$. Total computational time (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	59.5	61.7	67.6	98.6	133	225	377
CPTR(Block)	81.4	84.4	92.6	127	182	300	463
CPTR(uAMG)	41.6	45.7	53.5	95.2	202	382	659

Table 6.17: Weak scaling: 3D Heating case, $S_o = 0.5$. Total linear iterations.

Method/Num. proc.	1	2	4	8	16	32	64
CPR	103	140	172	197	231	288	324
CPTR(Block)	92	95	95	103	111	114	112
CPTR(uAMG)	98	99	102	116	123	125	126

Table 6.18: Weak scaling: 3D Heating case, $S_o = 0.5$. Average computational time per linear iteration (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	0.578	0.441	0.393	0.501	0.576	0.781	1.16
CPTR(Block)	0.885	0.888	0.975	1.24	1.64	2.63	4.13
CPTR(uAMG)	0.424	0.462	0.525	0.821	1.64	3.05	5.23

Table 6.19: Weak scaling: 3D Well case, $S_o = 0.99$. Total computational time (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	56.5	62.2	68.5	140	361	840	1710
CPTR(Block)	80.3	87.5	96.3	188	484	1060	2050
CPTR(uAMG)	40.2	45.9	52.8	148	464	1290	2760

We observe that, in both cases, CPR still takes less time than both versions of CPTR on the largest problem. For the block preconditioner, the first stage of CPTR is expected to be around 3 times more expensive than CPR to apply. The cost of applying the first stage of CPTR using uAMG should be around 2 times more

Table 6.20: Weak scaling: 3D Well case, $S_o = 0.99$. Total linear iterations.

Method/Num. proc.	1	2	4	8	16	32	64
CPR	73	101	121	246	565	1050	1550
CPTR(Block)	67	72	73	130	257	442	617
CPTR(uAMG)	67	72	75	140	260	446	626

Table 6.21: Weak scaling: 3D Well case, $S_o = 0.99$. Average computational time per linear iteration (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	0.774	0.616	0.566	0.570	0.638	0.8	1.11
CPTR(Block)	1.20	1.22	1.32	1.45	1.88	2.41	3.32
CPTR(uAMG)	0.6	0.638	0.704	1.06	1.78	2.89	4.41

expensive than for the first of CPR. Since we build a hierarchy for both pressure and temperature, setting up the first stage of CPTR should take twice as much time as for CPR. For each method, applying and setting up the second stage should be the same cost. At 64 cores, CPTR(Block) has a time per linear iteration 3 times greater than CPR, and is around 3.5 times more expensive for the heater case. CPTR(uAMG) is 4 and 4.5 times more expensive per iteration than CPR in the well and heater cases, respectively. The cost per iteration of CPTR(uAMG) seems to increase faster than for CPR and CPTR(Block). It is unclear why this is the case. Moreover, the reason for the observed increase in the cost per iteration in all 3D cases is unclear. GMRES is more expensive for later iterations, but there is only a mild increase in iteration count for CPTR in the heater case. Additionally, the cost of applying the preconditioners is more significant than the other costs associated with GMRES. Another explanation might be due to the methods being memory bandwidth-intensive. However, the cost increase appears even once the nodes are saturated (so results for 16, 32, and 64 cores). In [67], a study of the scalability of BoomerAMG compared to geometric multigrid is performed using Firedrake. They observe good scalability of both methods once the nodes are saturated.

We now consider a 2D well case similar to those in Section 6.2.3.1. We change the size of the domain to 20 meters $\times$ 20 meters. The $N \times N$ mesh is determined such that we have around 10,000 degrees of freedom per processor. We get $N = 58$, 82, 115, 163, 231, 327, and 462 for numbers of processors 1, 2, 4, 8, 16, 32, and 64, respectively. The production/injection rate is set to $q = 5 \times 10^{-8}$ $\mathrm{m}^3\,\mathrm{s}^{-1}$, and we simulate for 5 time-steps of size 5 days. The total computational time, total number

of linear iterations, and average computational time per linear iteration are shown in Tables 6.22 to 6.24. Since the iteration counts are larger here, both CPTR methods easily beat CPR. Most significantly, the cost per iteration does not vary as much here as it did for the 3D cases. Nonetheless, there is a larger increase for CPTR(uAMG).

Table 6.22: Weak scaling: 2D Small Well case, $S_o = 0.90$. Total computational time (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	2.13	2.55	2.86	3.92	7.85	11.8	36.3
CPTR(Block)	3.73	4.33	4.53	5.34	6.38	8.79	14.1
CPTR(uAMG)	2.18	2.52	2.65	3.64	4.11	5.12	11.5

Table 6.23: Weak scaling: 2D Small Well case, $S_o = 0.90$. Total number of linear iterations.

Method/Num. proc.	1	2	4	8	16	32	64
CPR	172	266	347	481	635	1010	3548
CPTR(Block)	86	96	92	107	110	131	319
CPTR(uAMG)	92	99	95	107	110	136	324

Table 6.24: Weak scaling: 2D Small Well case, $S_o = 0.90$. Average computational time per linear iteration (s).

Method/Num. proc.	1	2	4	8	16	32	64
CPR	0.0124	0.00959	0.00824	0.00815	0.0124	0.0117	0.0102
CPTR(Block)	0.0434	0.0451	0.0492	0.0499	0.058	0.0671	0.0442
CPTR(uAMG)	0.0237	0.0255	0.0279	0.0340	0.0374	0.0376	0.0355

Further study of the computational cost of these methods is desirable. With the current implementation, however, the limited profiling capabilities of the hypre library prevents an exhaustive study of how computational time is spent.

6.2.3.5 High pressure-temperature cross-coupling

In Section 6.2.3.1, we considered the use of unknown-based AMG (uAMG) and our block preconditioner for the first stage of CPTR. While uAMG usually results in more iterations, it exhibits similar scalability. However, uAMG is not expected to perform well under a strong cross-coupling. Here, we artificially increase the cross-coupling of the pressure-temperature system to compare the robustness of different

two-stage preconditioners. Along with uAMG and the block preconditioner, we also show results for CPTR where the first stage is a block diagonal preconditioner using either LU or AMG on the A_{pp} and A_{TT} blocks. To avoid confusion, we call these methods block-diagonal LU (bdLU), and block-diagonal AMG (bdAMG). We also consider CPR with AMG in the first stage.

Recall from Section 2.5 that we have only considered the flow of two fluids: water, and a heavy oil. The viscosity of the oil varies greatly with temperature, and the water viscosity much less so. As for their densities, liquid water and oil are much less compressible and thermally expansive than other fluids such as gases. To create a higher pressure-temperature cross-coupling, we increase the compressibility coefficient c and the thermal expansion coefficient β of the oil density as defined in (2.44).

We consider the 2D isotropic well case from Section 6.2.3.1 with $N = 160$. Both coefficients c and β are multiplied by the same increasing factor. Since larger coefficients result in a stiffer problem, smaller time-steps are needed for the convergence of Newton's method. Results for a time-step of 2 days and 0.1 days are shown in Tables 6.25 and 6.26, respectively. For CPTR, we observe that the block preconditioner exhibits a very small increase in iterations, while the other methods have a steady increase in iterations. It is clear that the block preconditioner is much better at dealing with a strong cross-coupling than the other versions of CPTR. For CPR, it shows similar robustness with respect to a strong cross-coupling. Note that by taking smaller time-steps, we are in a regime where CPR is competitive. Since CPR performs appropriately but CPTR (other than CPTR(Block)) does not, this may indicate that the coupling A_{Tp} is more important than A_{pT}. The temperature solution provided in the first stage of CPTR by solvers other than the block preconditioner is not accurate. It remains to be seen if A_{pT} is more important in other cases.

Table 6.25: Increased cross-coupling with $\Delta t = 2$ days. 2D Isotropic Well case. $N = 160$. Average linear iterations per nonlinear iteration.

Method/Factor	1	5	10	15	20
CPR	8.29	8.29	8.42	8.63	8.8
CPTR(Block)	5.29	5.29	5.57	6	6.27
CPTR(uAMG)	5.57	6.14	7	8.5	10.6
CPTR(bdLU)	8.29	9.14	10.1	11.1	12.8
CPTR(bdAMG)	8.29	9.14	10.3	11.1	12.8

In Table 6.26, the number of iterations eventually blows up for uAMG, bdLU, and bdAMG. However, in the case of uAMG, this may not necessarily be because of a

Table 6.26: Increased cross-coupling with $\Delta t = 0.1$ days. 2D Isotropic Well case. $N = 160$. Average linear iterations per nonlinear iteration.

Method/Factor	1	15	30	35	40
CPR	5.33	6.17	5	5.17	6.33
CPTR(Block)	5	5.83	4	4.33	5.17
CPTR(uAMG)	4.83	8.33	21.7	92.8	>200
CPTR(bdLU)	5.83	7.83	12.3	19.7	>200
CPTR(bdAMG)	6	7.5	12.3	19.7	>200

strong cross-coupling but perhaps that the properties of A_{TT} are no longer amenable to the application of AMG. Recall from Section 3.4.2 that the derivative of density with respect to temperature is negative. Consider the derivative of the oil enthalpy

$$(c_{v,o}\rho_o T)_T = c_{v,o}\rho_o + c_{v,o}(\rho_o)_T T, \tag{6.40}$$

which is positive for the default parameters of the earlier sections. However, for the larger factors in Table 6.26, the negative influence of $(\rho_o)_T$ begins to dominate. For the largest factors, this results in A_{TT} losing diagonal dominance, becoming indefinite, and eventually having a negative diagonal. This is, of course, far from the ideal properties for the convergence of AMG. For uAMG, the coarse pressure problem is coupled through temperature using the prolongation operator for the temperature coarse grid. Since that operator is not descriptive, this may indicate why uAMG is doing worse than bdAMG.

On the other hand, the block preconditioner does not apply AMG to A_{TT}, but rather to the approximate Schur complement $\tilde{S}_T$. Since this approximation does not include $(\rho_o)_T$, it will not suffer from a larger thermal expansivity coefficient.

6.3 Conclusion

For this chapter, we have implemented a fully implicit parallel non-isothermal multiphase flow in porous media simulator including two preconditioning strategies, CPR and CPTR. The first stage of CPTR can be a block preconditioner with our own Schur complement approximation, or an unknown-based AMG method. Standard decoupling operators are available for CPR, as well as their extension for CPTR.

On coarse grids, both CPR and CPTR exhibit a similar number of iterations, which means that CPR outperforms CPTR in terms of computational cost. In these cases, heat diffusion is not very significant, so CPTR is not necessary. In other cases,

however, CPTR displays much better scalability in terms of mesh refinement, as well as parallelization. Additionally, the first stage of CPTR is vulnerable to decoupling errors with saturation variables which can negate the scalability advantages of the method. Algebraic decoupling operators do not solve this issue. To deal with a strong pressure-temperature cross-coupling, our block preconditioner is better than the less coupled alternatives in the first stage of CPTR.

We believe that more work is needed to understand the decoupling error from the first stage of CPTR. To regain the scalability that CPTR displays in other cases, an accurate energy equation is key. It is still unclear if this can be done with another CPR-like multi-stage preconditioner.

Chapter 7

Conclusions

7.1 Summary

In this thesis, we have considered preconditioning strategies for the solution of non-isothermal flow in porous media. Our main focus has been to develop a two-stage preconditioner, the Constrained Pressure-Temperature Residual method (CPTR). To do so, we first needed an efficient pressure-temperature solver. By first considering single-phase flow, we designed a block preconditioner for the pressure-temperature system.

In Chapter 3, we discussed the use of decoupling operators with the Constrained Pressure Residual method (CPR). These perform an algebraic modification of the pressure system solved in the first stage of CPR. The main difficulty with decoupling operators is that the approximately decoupled pressure matrix has to be amenable to the application of Algebraic Multigrid (AMG). Even if the original pressure matrix is fully elliptic, the new matrix may not be. We determined that this was very sensitive to the alignment and choice of primary unknowns (those not eliminated via constraints). In the case of simple isothermal multiphase flow, this is easy to achieve. Without well terms, the Quasi-IMPES (QI) decoupling operator is robust under reasonable assumptions, while True-IMPES (TI) is automatically robust. For compositional flow, however, we easily show that certain choices can result in unpredictable properties of the pressure block. For a simple non-isothermal case, it is hard to confirm if decoupling operators lead to a good pressure block, although TI is observed to do so in practice.

In Chapter 4, we presented a weak formulation for the Finite Volume method typically used in reservoir simulation. This allows the use of the open source Finite Element software Firedrake [77], with which we implemented single-phase and two-

phase models. In turn, our preconditioning approaches can be implemented using Firedrake's interface with the solver library PETSc [9].

In Chapter 5, we considered different preconditioners for the single-phase case. We sought to design a good Schur complement approximation for an effective block preconditioner. By considering the differential operators associated with the different blocks of the system matrix, we constructed an approximation using heuristical arguments. Our Schur complement approximation performs significantly better than simple ones obtained algebraically. In terms of scalability with the problem size and parallelization, the block preconditioner performs much better than CPR. As opposed to CPR, the block preconditioner uses AMG to treat the heat diffusing and thus captures it appropriately. For advection-dominated cases, notably on very coarse grids, CPR is still very competitive.

In Chapter 6, we investigated preconditioning approaches for the multiphase case, and performed numerical tests for a two-phase model. The CPTR method is a CPR-like method where a restricted pressure-temperature system is approximately solved in the first stage. The Schur complement approximation presented in Chapter 5 is easily extended to the multiphase case. This leads to an effective block preconditioner for the first stage of CPTR. In most cases, CPTR exhibits good scaling properties like the block preconditioner for the single-phase case. However, in a highly heterogeneous test case on a coarse grid, as well as in a test case with a fully saturated reservoir, CPTR does not provide any advantage over CPR. In those cases, the displacement of the phases seems to be the main challenge for the solver rather than thermal effects. We then performed an investigation where a strong pressure-temperature cross-coupling is created artificially. CPR performed well under a strong coupling, which indicates that the coupling of the pressure equation with the temperature is not as important. For CPTR, our block preconditioner gave an appropriate solution of the pressure-temperature system, as opposed to less coupled methods which fail to do so. Decoupling operators were shown to perform decently when extended to CPTR. However, they are not compatible with the use of the block preconditioner.

A major issue with CPR in thermal reservoir simulation is its lack of treatment of heat diffusion. CPTR offers an alternative where heat diffusion is treated using AMG. Diffusion-dominated flows notably appear on fine meshes. Furthermore, since CPTR does not rely on ILU for the energy equation, it is often observed that CPTR has parallel scalability independent of ILU. However, when heat diffusion is not problematic for CPR, CPTR appears to offer little advantage.

We have briefly investigated strong pressure-temperature cross-couplings. For our limited test cases, we do not have systems where the coupling of the pressure equation with temperature is problematic. Such a scenario could be advantageous for CPTR with our block preconditioner since CPR ignores or only approximately decouples that coupling.

7.2 Future directions

While the models considered in this thesis are complex enough that they require advanced preconditioners, they do not include many of the features of the models in commercial reservoir simulators. Some of these features could be added to our models for further testing of CPTR with few modifications. More advanced models that include features such as poromechanics, electromagnetics (for heaters), or multiscale behaviour would require substantial effort and a different preconditioning approach.

For instance, even though we presented CPTR for an arbitrary number of phases, we have only tested two-phase flow. The addition of phases is an obvious next step in evaluating the performance of CPTR. Steam and hydrocarbon gases have very different properties than the fluids tested in this document. However, our experiment with artificially increased compressibility and thermal expansivity coefficients indicates that our Schur complement approximation could hold for gases. Mobility between phases is another challenging aspect of multiphase flow. CPTR could be useful in cases where phase changes are caused by temperature. In the case of compositional flow, the extension of our Schur complement approximation should be very similar to the multiphase case, i.e. with an appropriately constructed pressure equation.

Our numerical results show that our preconditioners perform well under uniform mesh refinements. It is more common in reservoir simulation to have mesh refinements solely around important features such as wells. Scalability in such cases would be industrially very relevant.

Additionally, the capillary pressure between fluids is an important feature of multiphase flow, which adds a diffusive effect to transport problems. In cases where the saturation problem is diffusion-dominated, applying AMG to the saturation block may be effective. This could also lead to interesting block preconditioners that do not rely on ILU.

Other applications than oil and gas recovery may benefit from our preconditioning approaches. For example, they could readily be applied to geothermal energy cases.

Alternatively, our block preconditioner could be modified for an isothermal transport problem where the molecular diffusion of a species is analogous to heat diffusion.

Our Firedrake implementation does not have ready access to all the test cases and client cases that a commercial reservoir simulator would. The wide range of parameters and possible configurations of thermal cases would give a better idea of the when CPTR is needed over CPR. Our tests only looked at algorithmic performance on a small scale. Further testing of larger problems on larger architectures with a comparison of computational time would also be very useful.

Appendix A

Definitions

Definition 1. A matrix is *diagonally dominant* if the coefficients on its diagonal are greater than the sums of the absolute values of the off-diagonal coefficients of their respective rows, i.e. for a matrix A,

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \qquad \text{for all } i. \tag{A.1}$$

Similarly, a matrix is said to be *strictly diagonally dominant* if the inequality in (A.1) is strict.

Definition 2. A matrix is a *Z-matrix* if its off-diagonal entries are non-positive. A Z-matrix A is an *M-matrix* if its eigenvalues have positive real parts.

Equivalently a Z-matrix A is an M-matrix if all its diagonal entries are positive and there exists a positive diagonal matrix D such that AD is strictly diagonally dominant.

Definition 3. A real square matrix A is *monotone* if for all real vectors x, all elements of Ax being non-negative implies that all elements of x are non-negative.

Definition 4. A *non-negative* matrix is such that all of its elements are non-negative.

A matrix is monotone if and only if its inverse is non-negative.

Appendix B

Decoupling Coefficients

Let $u^\top = \begin{pmatrix} x & y \end{pmatrix}$ with $x, y > 0$, and

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \tag{B.1}$$

be nonsingular with $a > 0$, $d > 0$.

Proposition 1. The following conditions are sufficient for $u^\top A^{-1}$ to have non-negative entries.

(i) $c < 0$, $ad - bc > 0$, and $bx \leq ay$;

(ii) $c \geq 0$, $cy < dx$, and $bx \leq ay$;

(iii) $cy > dx$, and $bx \geq ay$.

Proof. The inverse of A is given by

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}. \tag{B.2}$$

Then $u^\top A^{-1} = \frac{1}{ad-bc} \begin{pmatrix} dx - cy & ay - bx \end{pmatrix}$. The results easily follow. □

In particular, if $x = y$, we only require A to have diagonal dominance. The condition $dx \geq cy$, $ay \geq bx$ essentially means that A requires a type of diagonal dominance in the sense that

$$AD = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} y & 0 \\ 0 & x \end{pmatrix}, \tag{B.3}$$

is diagonally dominant. Also, if A is an M-matrix, it has $b, c < 0$ and a positive determinant so it easily satisfies condition (i).

Proposition 2. The following conditions are sufficient for the vector given by $(u^\top A^\top)(AA^\top)^{-1}$ to have non-negative entries:

(i) A is diagonally dominant, and $b, c \geq 0$;

(ii) A is monotone, $ax + by \geq 0$, and $cx + dy \geq 0$;

(iii) A is an M-matrix, $d^2(ax+by)-bc(cx+dy) \geq 0$, and $a^2(cx+dy)-bc(ax+by) \geq 0$.

Proof. We have that

$$A^\top (AA^\top)^{-1} = \frac{1}{a^2d^2 - b^2c^2} \begin{pmatrix} ad^2 - bc^2 & ac(a-b) \\ bd(d-c) & a^2d - b^2c \end{pmatrix}. \tag{B.4}$$

If condition (i) holds, the result easily follows since $a > b$ and $d > c$. Also, $u^\top A^\top = \begin{pmatrix} ax + by & cx + dy \end{pmatrix}$.

A being monotone implies that $(AA^\top)^{-1}$ is a non-negative matrix and so condition (ii) implies the result.

We have that

$$u^\top A^\top (AA^\top)^{-1} = \frac{1}{a^2d^2 - b^2c^2} \begin{pmatrix} d^2(ax + by) - bc(cx + dy) & a^2(cx + dy) - bc(ax + by) \end{pmatrix}. \tag{B.5}$$

If condition (iii) holds, A has a positive determinant and so $ad > bc$, and thus the result follows. □

Bibliography

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

[2] G. Acs, S. Doleschall, and E. Farkas. General purpose compositional model. *Society of Petroleum Engineers Journal*, 25(04):543–553, 1985.

[3] T. M. Al-Shaalan, H. M. Klie, A. H. Dogru, and M. F. Wheeler. Studies of robust two stage preconditioners for the solution of fully implicit multiphase flow problems. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009.

[4] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[5] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language: a domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 40(2):9, 2014.

[6] J. R. Appleyard. Nested factorization. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1983.

[7] K. Aziz and A. Settari. *Petroleum reservoir simulation*. Applied Science Publishers, 1979.

[8] Z.-Z. Bai and M. K. Ng. On inexact preconditioners for nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 26(5):1710–1724, 2005.

[9] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page, 2017. URL: `http://www.mcs.anl.gov/petsc`.

[10] R. E. Bank, T. F. Chan, W. M. Coughran Jr, and R. K. Smith. The alternate-block-factorization procedure for systems of partial differential equations. *BIT Numerical Mathematics*, 29(4):938–954, 1989.

[11] T. Bennison. Prediction of heavy oil viscosity. In *Presented at the IBC Heavy Oil Field Development Conference*, volume 2, page 4, 1998.

[12] R. Booth. *Miscible flow through porous media*. DPhil thesis, University of Oxford, 2008.

[13] G. Boyle. *Renewable Energy: Power for a Sustainable Future*. Oxford University Press, 2012, page 584.

[14] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[15] F. Brezzi, T. J. R. Hughes, L. D. Marini, and A. Masud. Mixed discontinuous Galerkin methods for Darcy flow. *Journal of Scientific Computing*, 22(1-3):119–145, 2005.

[16] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. SIAM, 2000.

[17] Q. M. Bui, H. C. Elman, and J. D. Moulton. Algebraic multigrid preconditioners for multiphase flow in porous media. *SIAM Journal on Scientific Computing*, 39(5):S662–S680, 2017.

[18] Q. M. Bui, L. Wang, and D. Osei-Kuffuor. Algebraic multigrid preconditioners for two-phase flow in porous media with phase transitions. *Advances in water resources*, 114:19–28, 2018.

[19] R. M. Butler. *Thermal recovery of oil and bitumen*, volume 46. Prentice Hall Englewood Cliffs, NJ, 1991.

[20] H. Cao, H. A. Tchelepi, J. R. Wallis, and H. E. Yardumian. Parallel scalable unstructured CPR-type linear solver for reservoir simulation. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2005.

[21] N. Castelletto, J. A. White, and M. Ferronato. Scalable algorithms for three-field mixed finite element coupled poromechanics. *Journal of Computational Physics*, 327:894–918, 2016.

[22] Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media*. SIAM, 2006.

[23] Z. Chen and Y. Zhang. Well flow models for various numerical methods. *International Journal of Numerical Analysis & Modeling*, 6(3), 2009.

[24] E. Chow and A. Patel. Fine-grained parallel incomplete LU factorization. *SIAM journal on Scientific Computing*, 37(2):C169–C193, 2015.

[25] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: a comparison of upscaling techniques. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2001.

[26] T. Clees. *AMG strategies for PDE systems with applications in industrial semiconductor simulation*. PhD thesis, Universität zu Köln, 2005.

[27] T. Clees and L. Ganzer. An efficient algebraic multigrid solver strategy for adaptive implicit methods in oil-reservoir simulation. *Society of Petroleum Engineers Journal*, 15(03):670–681, 2010.

[28] K. H. Coats. In-situ combustion model. *Society of Petroleum Engineers Journal*, 20(06):533–554, 1980.

[29] H. Darcy. *Les fontaines publiques de la ville de Dijon*. Victor Dalmont, 1856.

[30] D. DeBaun, T. Byer, P. Childs, J. Chen, F. Saaf, M. Wells, J. Liu, H. Cao, L. Pianelo, V. Tilakraj, P. Crumpton, D. Walsh, H. Yardumian, R. Zorzynski, K.-T. Lim, M. Schrader, V. Zapata, J. Nolen, and H. Tchelepi. An extensible architecture for next generation scalable parallel reservoir simulation. In *SPE Reservoir Simulation Symposium*, 2005.

[31] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation*. SIAM, 2015.

[32] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, USA, 2014.

[33] M. Embree. How descriptive are GMRES convergence bounds? Technical report, Oxford University Computing Laboratory, 1999.

[34] Energy Education. Steam assisted gravity drainage. 2018. URL: `http://energyeducation.ca/encyclopedia/Steam_assisted_gravity_drainage` (visited on 09/19/2019).

[35] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. *Handbook of Numerical Analysis*, 7:713–1018, 2000.

[36] R. D. Falgout and U. M. Yang. Hypre: a library of high performance preconditioners. In *International Conference on Computational Science*, pages 632–641. Springer, 2002.

[37] C. R. Faust and J. W. Mercer. Geothermal reservoir simulation: 1. mathematical models for liquid-and vapor-dominated hydrothermal systems. *Water resources research*, 15(1):23–30, 1979.

[38] L. S. K. Fung and A. H. Dogru. Parallel unstructured-solver methods for simulation of complex giant reservoirs. *Society of Petroleum Engineers Journal*, 13(04):440–446, 2008.

[39] M. W. Gee, C. M. Siefert, J. J. Hu, R. S. Tuminaro, and M. G. Sala. ML 5.0 Smoothed Aggregation User's Guide. Technical report SAND2006-2649, Sandia National Laboratories, 2006.

[40] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.

[41] J. W. Grabowski, P. K. Vinsome, R. C. Lin, G. Behie, and B. Rubin. A fully implicit general purpose finite-difference thermal model for in situ combustion and steam. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 1979.

[42] S. Gries. *System-AMG Approaches for Industrial Fully and Adaptive Implicit Oil Reservoir Simulations*. PhD thesis, Universität zu Köln, 2015.

[43] S. Gries, B. Metsch, K. M. Terekhov, and P. Tomin. System-AMG for fully coupled reservoir simulation with geomechanics. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2019.

[44] S. Gries, K. Stüben, G. L. Brown, D. Chen, and D. A. Collins. Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations. *Society of Petroleum Engineers Journal*, 19(04):726–736, 2014.

[45] W. Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20(4):291–306, 1978.

[46] V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 2002.

[47] R. M. Holland, A. J. Wathen, and G. J. Shaw. Sparse approximate inverses and target matrices. *SIAM Journal on Scientific Computing*, 26(3):1000–1011, 2005.

[48] I. C. F. Ipsen. A note on preconditioning nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 23(3):1050–1051, 2001.

[49] S. Jauré, A. Moncorgé, and R. de Loubens. Reservoir simulation prototyping platform for high performance computing. In *SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference and Exhibition*. Society of Petroleum Engineers, 2014.

[50] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 24(1):237–256, 2002.

[51] G. S. Kell. Density, thermal expansivity, and compressibility of liquid water from 0. deg. to 150. deg.. correlations and tables for atmospheric pressure and saturation reviewed and expressed on 1968 temperature scale. *Journal of Chemical and Engineering Data*, 20(1):97–105, 1975.

[52] R. C. Kirby and L. Mitchell. Solver composition across the PDE/linear algebra barrier. *SIAM Journal on Scientific Computing*, 40(1):C76–C98, 2018.

[53] H. Klie, M. Rame, and M. Wheeler. Two-stage preconditions for inexact Newton methods in multi-phase reservoir simulation. Technical report CRPC-TR96641-S, Center for Research on Parallel Computation, Rice University, 1996.

[54] S. Lacroix, Y. V. Vassilevski, J. Wheeler, and M. F. Wheeler. Iterative solution methods for modeling multiphase flow in porous media fully implicitly. *SIAM Journal on Scientific Computing*, 25(3):905–926, 2003.

[55] S. Lacroix, Y. V. Vassilevski, and M. F. Wheeler. Iterative solvers of the implicit parallel accurate reservoir simulator (IPARS), I: single processor case. TICAM report 00-28, The University of Texas at Austin, 2000.

[56] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge University Press, 2002.

[57] G. Li and J. Wallis. Enhanced constrained pressure residual ECPR preconditioning for solving difficult large scale thermal models. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[58] G. Li, J. Wallis, and G. Shaw. A parallel linear solver algorithm for solving difficult large scale thermal models. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2015.

[59] A. Logg, K.-A. Mardal, and G. N. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[60] R. C. MacDonald and K. H. Coats. Methods for numerical simulation of water and gas coning. *Transactions of the AIME*, 249:425–436, 1970.

[61] Y. Maday, D. Meiron, A. T. Patera, and E. M. Rønquist. Analysis of iterative methods for the steady and unsteady Stokes problem: application to spectral element discretizations. *SIAM Journal on Scientific Computing*, 14(2):310–337, 1993.

[62] K.-A. Mardal and R. Winther. Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications*, 18(1):1–40, 2011.

[63] J. W. Marx and R. H. Langenheim. Reservoir heating by hot fluid injection. *Transactions of the AIME*, 216:312–315, 1959.

[64] A. Masud and T. J. R. Hughes. A stabilized mixed finite element method for Darcy flow. *Computer methods in applied mechanics and engineering*, 191(39-40):4341–4370, 2002.

[65] Q. Mehmood. Is there oil beneath my property? 2015. URL: http://geologylearn.blogspot.co.uk/2015/08/is-there-oil-beneath-my-property-first.html (visited on 09/19/2019).

[66] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31(137):148–162, 1977.

[67] L. Mitchell and E. H. Müller. High level implementation of geometric multigrid solvers for finite element problems: applications in atmospheric modelling. *Journal of Computational Physics*, 327:1–18, 2016.

[68] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21(6):1969–1972, 2000.

[69] M. Muskat and M. W. Meres. The flow of heterogeneous fluids through porous media. *Physics*, 7(9):346–363, 1936.

[70] Y. Notay. A robust algebraic multilevel preconditioner for non-symmetric M-matrices. *Numerical linear algebra with applications*, 7(5):243–267, 2000.

[71] D. S. Oliver and Y. Chen. Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15(1):185–221, 2011.

[72] A.-T. Papadopoulos. *Block smoothed aggregation algebraic multigrid preconditioners for oil reservoir simulation systems*. DPhil thesis, University of Oxford, 2004.

[73] S. V. Patankar. *Numerical heat transfer and fluid flow*. Series on Computational Methods in Mechanics and Thermal Science. Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group), 1980.

[74] D. W. Peaceman. *Fundamentals of numerical reservoir simulation*. Elsevier, 1977.

[75] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988). *Society of Petroleum Engineers Journal*, 18(03):183–194, 1978.

[76] C. Qiao, S. Wu, J. Xu, and C.-S. Zhang. Analytical decoupling techniques for fully implicit reservoir simulation. *Journal of Computational Physics*, 336:664–681, 2017.

[77] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):24, 2016.

[78] M. M. Rehman and M. Meribout. Conventional versus electrical enhanced oil recovery: a review. *Journal of Petroleum Exploration and Production Technology*, 2(4):157–167, 2012.

[79] A. N. Riseth. Nonlinear solver techniques in reservoir simulation. Technical report, Oxford University Mathematical Institute, 2015.

[80] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. SIAM, 2008.

[81] T. Roy. Improved Algebraic Decoupling of Pressure Equation in Reservoir Simulation. Technical report, Oxford University Mathematical Institute, 2016.

[82] T. Roy, T. B. Jönsthövel, C. Lemon, and A. J. Wathen. A block preconditioner for non-isothermal flow in porous media. *Journal of Computational Physics*, 395:636–652, 2019.

[83] T. Roy, T. B. Jönsthövel, C. Lemon, and A. J. Wathen. A constrained pressure-temperature residual (CPTR) method for non-isothermal multiphase flow in porous media. *arXiv preprint arXiv:1907.04229*, 2019.

[84] J. W. Ruge. AMG for problems of elasticity. *Applied Mathematics and Computation*, 19(1-4):293–309, 1986.

[85] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*. Volume 3 of Frontiers in Applied Mathematics, chapter 4, pages 73–130. SIAM, Philadelphia, 1987.

[86] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[87] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[88] P. G. Saffman and G. I. Taylor. The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 245(1242):312–329, 1958.

[89] A. Sahni, M. Kumar, and R. B. Knapp. Electromagnetic heating methods for heavy oil reservoirs. In *SPE/AAPG Western Regional Meeting*. Society of Petroleum Engineers, 2000.

[90] R. Scheichl, R. Masson, and J. Wendebourg. Decoupling and block preconditioning for sedimentary basin simulations. *Computational Geosciences*, 7(4):295–318, 2003.

[91] J. W. Sheldon and W. T. Cardwell Jr. One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium. *Transactions of the AIME*, 216:290–296, 1959.

[92] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31(5):1352–1367, 1994.

[93] G. L. Sleijpen and D. R. Fokkema. BiCGSTAB(l) for linear equations involving unsymmetric matrices with complex spectrum. *Electronic Transactions on Numerical Analysis*, 1:11–32, 1993.

[94] A. G. Spillette, J. G. Hillestad, and H. L. Stone. A high-stability sequential solution approach to reservoir simulation. In *Fall Meeting of the Society of Petroleum Engineers of AIME*. Society of Petroleum Engineers, 1973.

[95] H. L. Stone and A. O. Garder Jr. Analysis of gas-cap or dissolved-gas drive reservoirs. *Society of Petroleum Engineers Journal*, 1(02):92–104, 1961.

[96] K. Stüben. A review of algebraic multigrid. In *Numerical Analysis: Historical Developments in the 20th Century*, pages 331–359. Elsevier, 2001.

[97] K. Stüben. Algebraic multigrid (AMG): an introduction with applications. Technical report GMD-Report 70, Institute for Algorithms and Scientific Computing (SCAI), 1999.

[98] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. W. Oosterlee, and A. Schuller, editors, *Multigrid*, pages 413–532. Academic Press, 2001.

[99] K. Stüben, T. Clees, H. Klie, B. Lu, and M. F. Wheeler. Algebraic multigrid methods (AMG) for the efficient solution of fully implicit formulations in reservoir simulation. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2007.

[100] G. W. Thomas and D. H. Thurnau. Reservoir simulation using an adaptive implicit method. *Society of Petroleum Engineers Journal*, 23(05):759–768, 1983.

[101] J. A. Trangenstein and J. B. Bell. Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM Journal on Applied Mathematics*, 49(3):749–783, 1989.

[102] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic press, 2000.

[103] H. A. Van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[104] D. E. Van Odyck, J. B. Bell, F. Monmont, and N. Nikiforakis. The mathematical structure of multiphase thermal models of flow in porous media. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 465 of number 2102, pages 523–549. The Royal Society, 2009.

[105] J. R. Wallis. Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1983.

[106] J. R. Wallis, R. P. Kendall, and T. E. Little. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1985.

[107] L. Wang, D. Osei-Kuffuor, R. Falgout, I. Mishev, and J. Li. Multigrid reduction for coupled flow problems with application to reservoir simulation. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[108] A. J. Wathen. Preconditioning. *Acta Numerica*, 24:329–376, 2015.

[109] J. W. Watts. A compositional formulation of the pressure and saturation equations. *SPE Reservoir Engineering*, 1(03):243–252, 1986.

[110] S. Whitaker. Flow in porous media I: a theoretical derivation of Darcy's law. *Transport in porous media*, 1(1):3–25, 1986.

[111] J. A. White, N. Castelletto, S. Klevtsov, Q. M. Bui, D. Osei-Kuffuor, and H. A. Tchelepi. A two-stage preconditioner for multiphase poromechanics in reservoir simulation. *Computer Methods in Applied Mechanics and Engineering*, 357:112575, 2019.

[112] J. A. White, N. Castelletto, and H. A. Tchelepi. Block-partitioned solvers for coupled poromechanics: a unified framework. *Computer Methods in Applied Mechanics and Engineering*, 303:55–74, 2016.

[113] Z. Y. Wong. *Sequential-implicit Newton's Method for Geothermal Reservoir Simulation*. PhD thesis, Stanford University, 2018.