ABSTRACT

Title of dissertation: FAST SOLVERS AND PRECONDITIONERS FOR MULTIPHASE FLOW IN POROUS MEDIA

Quan M. Bui
Doctor of Philosophy, 2018

Dissertation directed by: Professor Howard Elman
Department of Computer Science

Multiphase flow is a critical process in a wide range of applications, including carbon sequestration, contaminant remediation, and groundwater management. Typically, this process is modeled by a nonlinear system of partial differential equations derived by considering the mass conservation of each phase (e.g., oil, water), along with constitutive laws for the relationship of phase velocity to phase pressure. The problem becomes much more complex if the phases are allowed to contain multiple chemical species (also called components), as miscibility and phase transition effects need to be taken into account. The main problem with phase transition stems from the inconsistency of the primary variables such as phase pressure and phase saturation, i.e. they become ill-defined when a phase appears or disappears. Recently, a new approach for handling phase transition has been developed by formulating the system as a nonlinear complementarity problem (NCP). Unlike the widely used

primary variable switching method (PVS), which requires a drastic reduction of the time step size when a phase appears or disappears, this approach is more robust and allows for larger time steps. One way to solve an NCP system is to reformulate the inequality constraints for the primary variables as a non-smooth equation using a complementary function (C-function). Because of the non-smoothness of the constraint equations, a semi-smooth Newton method needs to be developed. Another feature of the NCP approach is that the set of primary variables in this approach is fixed even when there is phase transition. Not only does this improve the robustness of the nonlinear solver, it opens up the possibility to use multigrid methods to solve the resulting linear system. The disadvantage of the complementarity approach, however, is that when a phase disappears, the linear system has the structure of a saddle point problem and becomes indefinite, and current algebraic multigrid (AMG) algorithms cannot be applied directly.

In this work, we aim to address computational issues related to modeling multiphase flow in porous media. First, we develop and study efficient solution algorithms for solving the algebraic systems of equations derived from a fully coupled and time-implicit treatment of models of incompressible two-phase flow. We explore the performance of several preconditioners based on algebraic multigrid (AMG) for solving the linearized problem, including "black-box" AMG applied directly to the system, a new version of constrained pressure residual multigrid (CPR-AMG) preconditioning, and a new preconditioner derived using an approximate Schur complement arising from the block factorization of the Jacobian. We show that the new methods are the most robust with respect to problem character as determined

by varying effects of capillary pressures, and we show that the block factorization preconditioner is both efficient and scales optimally with problem size. We then generalize the block factorization method and incorporate it into a multigrid framework which is based on the multigrid reduction technique to deal with linear systems resulting from the NCP approach for modeling compositional multiphase flow with phase transitions. We demonstrate the effectiveness and scalability of the method through numerical results for a case of two-phase, two-component flow with phase appearance/disappearance. Finally, we propose a new semi-smooth Newton method which employs a smooth version of the Fischer-Burmeister function as the C-function and evaluate its performance against the semi-smooth Newton method for two C-functions: the minimum and the Fischer-Burmeister functions. We show that the new method is robust and efficient for standard benchmark problems as well as for realistic examples with highly heterogeneous media such as the SPE10 benchmark.

FAST SOLVERS AND PRECONDITIONERS FOR MULTIPHASE FLOW IN POROUS MEDIA

by

Quan M. Bui

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Howard Elman, Chair/Advisor
Professor Konstantina Trivisa
Professor Amir Riaz
Professor Kayo Ide
Professor Thomas Goldstein

Table of Contents

List of Tables

List of Figures

Abbreviations

AMG algebraic multigrid.

AMR adaptive mesh refinement.

ASCEM Advanced Simulation Capability for Environmental Management.

BF block factorization.

BiCGStab biconjugate gradient with stabilization.

Block FSAI Block Factorized Sparse Approximate Inverse.

CFL Courant-Friedrichs-Lewy condition.

CPR-AMG constrained pressure residual multigrid.

GMRES generalized residual method.

ILU incomplete LU factorization.

IMPES implicit pressure explicit saturation.

MFD mimetic finite difference.

MGR multigrid reduction.

NCP nonlinear complementarity problem.

PDEs partial differential equations.

PVS primary variable switching.

REV representative elementary volume.

SS simultaneous solution.

TPFA two-point flux-approximation.

Nomenclature

$\boldsymbol{q}_\alpha$ Darcy velocity of phase α.

$\boldsymbol{q}$ Total Darcy velocity.

ρ_α Density of phase α.

ρ_α^k Density of component k in phase α.

ψ^k Flux term of component k.

$\boldsymbol{F}_\alpha$ Forcing term of phase α.

D_α^k Molecular diffusion of component k in phase α.

f_α Fractional flow of phase α.

X_α^k Mass fraction of component k in phase α.

C_α^k Volume fraction of component k in phase α.

$\boldsymbol{g}$ Gravity vector.

$\boldsymbol{G}$ Modified gravity vector.

H Henry's law constant.

R Ideal gas constant.

λ_α Mobility of phase α.

λ Total mobility.

M^k Molar mass of component k.

$\boldsymbol{K}$ Absolute permeability.

$k_{r\alpha}$ Relative permeability of phase α.

ϕ Porosity.

P_α Pressure of phase α.

P_c Capillary pressure.

π_n Complementary pressure.

P_r Entry pressure.

P Global pressure.

S_α Saturation of phase α.

$S_{\alpha e}$ Effective saturation of phase α.

$S_{\alpha r}$ Residual saturation of phase α.

ξ^k Storage term of component k.

T Temperature.

μ_α Viscosity of phase α.

Chapter 1: Introduction

This dissertation is concerned with various computational issues that arise in modeling multiphase flow in porous media. In particular, we focus on developing fast algorithms for solving the algebraic systems resulting from the discretization of the fully coupled partial differential equations (PDEs) and constitutive laws that describe multiphase flow. The main contribution of this work includes robust solvers and preconditioners for complex models of multiphase, multicomponent flow that take into account important physical processes such as capillary pressure, miscibility, and phase transitions.

1.1 Overview of Multiphase Flow in Porous Media

Modeling multiphase flow in porous media is a subject of great complexity with a long and rich history in the field of fluid mechanics. The earliest applications of multiphase flow appear in soil science literature, in which an unsaturated flow consisted of water and air occurs in the soil [77]. In the 1980s, interest in the topic dramatically increased due to the rise in demand for flow simulation in petroleum engineering. Natural oil reservoirs almost always contain connate water and they sometimes contain free natural gas as well. Thus, engineers needed to understand the simultaneous flow of water, oil, and gas through porous media to maximize the amount of oil and gas extracted. More recently, the study of multiphase flows has garnered interest from hydrologists whose work involves groundwater quality management [4,18]. More and more contaminants are being released into the environment, posing a grave threat to underground water resources. These oily liquids, coming from underground or near-surface storage facilities, land-fills at which chemical wastes are dumped, industrial sites such as oil refineries or wood-treatment plants, or illegal waste disposal, may enter the water-bearing rock formations as separate, non-aqueous phases. Consequently, the ability to develop accurate multiphase flow models is critical to the design of sound remedial measures. Another application in which multiphase flow has been used extensively recently is carbon sequestration [30]. In order to slow down accumulation of greenhouse gases released from burned fossil fuels, much research proposes capturing and storing carbon dioxide and other forms of carbon in underground aquifers and depleted oil fields. These carbon products may be injected into a carbon sink either in their gaseous or liquefied state, or as a mixture of both. In all of these applications, the inherent complexity of the physics being modeled gives rise to highly nonlinear systems of PDEs, whose difficulty is further enhanced by the irregularities in the geometry and heterogeneity of the porous medium. Hence, numerical solution is the only effective and feasible way to gain quantitative insights into the behavior of multiphase flow in porous media.

1.2 Numerical Methods for Modeling Multiphase Flow in Porous Media

The system of PDEs modeling multiphase flow in porous media can be numerically approximated by a discretization process in space and time. For the spatial component, the most popular methods for multiphase flow equations are finite element and finite volume methods. In this work, we implement a finite volume method with two-point flux approximation. The resulting discrete nonlinear algebraic system is subsequently solved using a Newton-type method. We consider a standard Newton's method for the case of incompressible two-phase flow, and a family of semi-smooth Newton methods for compositional two-phase flow with phase transitions.

Each iteration of the nonlinear solve requires a linearization of the problem. This process also entails finding solutions to a series of large linear systems. In general, these systems, embedded with the strong physical coupling between the variables in the original PDEs, are sparse, non-symmetric, and highly indefinite. Since a large percentage of the computation in multiphase flow simulation is spent on solving these linear systems, efficient solution methods are critical to obtaining simulation results in a short amount of time. Scalability is also of the utmost importance. As scientists constantly demand simulations for bigger problems with ever-higher levels of details, designing algorithms that can take advantage of the massive parallelism that emerging architectures provide remains one of the few viable ways of moving forward.

1.2.1 Decoupled Approach

One way to reduce the complexity of the system of equations for multiphase flow in porous media is the separation of different physical processes involved. For example, in two-phase flow one can manipulate the conservation equation for each phase to obtain two separate equations: one for pressure and one for saturation. The pressure equation expresses the distribution of the pressure and it is usually governed by a diffusion process. In contrast, the saturation equation models a transport mechanism, having the characteristics of a mixed problem, i.e. advection-diffusion. With this approach, instead of solving a big nonlinear system of PDEs, one instead needs to solve two separate linear PDEs that complement each other, that is the solution of one equation serves as the input for the other. For example, given a distribution of the saturation, one can first solve the pressure equation to get a new pressure field, which is then substituted into the saturation equation. The saturation is subsequently obtained by solving the saturation equation with the updated pressure field. This iterating process between the two equations is repeated until convergence. The decoupling approach has the advantages that the linear systems associated with each equation are smaller, and they are each derived from a single physical process for which efficient solvers exist. Thus, for mildly nonlinear problems in which the coupling between pressure and saturation is weak, it could work well. However, when one takes into account important effects present in multiphase flow, including capilarity, miscibility, and phase transitions, the coupling between the variables becomes much stronger, and the decoupling approach loses its effectiveness. By decomposing the fully coupled multiphase flow system into its component parts, one also introduces splitting errors, and it may take many iterations between the component equations (pressure and saturation equations in the example above) to achieve a desired accuracy. In many cases, the process may not converge at all. Thus, for more sophisticated models of multiphase flow, a fully coupled approach is usually preferred.

1.2.2 Fully Coupled Approach

In a fully coupled approach, also called the simultaneous solution (SS) method in the reservoir simulation community, the system governing multiphase flow is discretized, linearized and solved together as one big system. Due to the size of the problem, Krylov subspace methods are a natural choice of solver. Yet, these methods, including generalized residual method (GMRES) [84] and biconjugate gradient with stabilization (BiCGStab) [90], cannot be used as standalone solvers. To obtain fast convergence, they need to be paired with robust and efficient preconditioners. The challenge of the fully coupled solution strategy is then to develop preconditioners that are capable of capturing the complex physical processes embedded in the discretized multiphase flow system, and resolving the strong coupling between the variables such as pressure, saturation, and concentration at the same time. Given a good preconditioner, this strategy is very efficient, as only one linear solve is required for each iteration of the nonlinear solve. Overall, the speed of the algorithm depends on the rate of convergence of the nonlinear solver and the cost of the linear solves.

1.3 The Problem with Phase Transitions

For compositional multiphase flow, handling phase transitions is a challenging task. When formulating the set of governing equations, most common approaches use the constraints for local equilibrium to eliminate some of the unknowns to get a reduced set of primary variables. For example, in a two-phase flow involving gas and liquid phases, using the constraint that the gas and liquid saturations sum to 1, we can eliminate the gas saturation and choose the liquid saturation as a primary variable. Problems arise when the liquid phase disappears. In such a situation, the reduced set of equations consists of variables that are not well defined, i.e. liquid saturation and liquid pressure are physically meaningless if there is no liquid phase present. As a consequence, the corresponding linear systems become singular and uniqueness of the solution is no longer guaranteed.

The classical and immensely popular way to overcome this problem is the primary variable switching (PVS) approach [51,98]. As the name suggests, the primary variables may be switched depending on the conditions of the phases. For example, in the regions where the gas phase turns into the liquid phase and disappears completely, one uses the system of equations and variables associated with the liquid phase even if the original system of equations and primary variables are those associated with the gas phase. Despite its success in handling phase transitions, the PVS approach has two main drawbacks: it exhibits irregular global convergence behavior in the nonlinear solves, and it relies on inefficient serial algorithms such as incomplete LU factorization (ILU) [83] for linear solves. A scalable solver such as multigrid [86, 88], however, cannot be applied directly without extensive modification for PVS. Thus, our goal is to find a robust algorithm that can seamlessly handle phase transitions that is also more readily adapted to multilevel solvers. Besides PVS, other approaches have been developed to address phase transitions. For example, one can introduce a set of so-called persistent primary variables that are well defined across all the phases (see [18, 63, 65]). This approach is limited in the sense that such persistent sets of primary variables may not exist for complicated compositional multiphase flow problems. Another approach is the method of negative saturation in [1], in which the saturation of a phase is allowed to be negative or greater than 1. Using this approach, one needs to use a post-processing step to recover a meaningful value for the solution when phase transitions occur. Since there are questions on how this can be done appropriately, this approach has not been widely adopted. A more recent approach is to formulate the system governing compositional two-phase flow with phase transitions as a nonlinear complementarity problem (NCP) [16, 59, 63]. The NCP formulation utilizes the fact that the variables of interest, e.g. pressure, saturation, concentrations, etc., must satisfy certain constraints to be physically meaningful when phase transitions occur. By explicitly enforcing these constraints, the solution to an NCP also satisfies the phase transition conditions. Due to its novelty and robustness in handling phase transitions, this is the method we employ in this work.

1.4 Outline of Dissertation

The outline of this dissertation is as follows. Chapter 2 introduces the set of equations modeling multiphase, multicomponent flow in porous media. We review the existing approaches for dealing with phase transitions and solution strategies, including the decoupled and fully coupled approaches. We then present a fully implicit discretization using a finite volume method in space and the backward Euler method in time for the fully coupled approach. The last section of this chapter gives an overview of common preconditioning techniques for the linear systems resulting from discretization of the equations modeling multiphase, multicomponent flow.

Chapters 3 and 4 focus on the development of efficient and scalable preconditioners for solving the linear systems resulting from a fully implicit discretization of incompressible two-phase flow, and compositional two-phase flow with phase transitions. A block factorization preconditioner is presented in chapter 3. In chapter 4, we formulate the block factorization preconditioner as a multigrid reduction approach and extend it to solve the compositional multiphase flow system with phase transitions. Chapter 5 deals with issues encountered in the nonlinear solve for the NCP reformulation of compositional two-phase flow with phase transitions. We develop a smoothing approach that is more efficient and robust in handling phase transitions than the current semi-smooth Newton methods. Finally, we present some concluding remarks in chapter 6.

Chapter 2: Problem Statement and Survey of Existing Approaches

In the first part of this chapter, a system of partial differential equations describing two-phase flow in a porous medium is derived. Before this derivation can be accomplished, we provide definitions and conceptual terms that are used in modeling flow in porous media. The subsequent sections are devoted to the current development of quantitative models for incompressible two-phase flow and compositional two-phase flow.

2.1 Conceptual Model of Flow in Porous Media

2.1.1 Definitions

A porous medium can be defined as any material consisting of two parts: the solid frame, also called *solid matrix*, and the *void space* (or *pore space*), which can be filled by one or more fluids (e.g. water, oil, gas, etc.). Many natural substances ranging from geological formation (rocks, soils, sand, limestone, etc.) to biological tissues (bones, wood, kidney) and man made materials (cements, ceramics) can be considered porous media.

According to [11], a *phase* is a chemically consistent body of a single fluid (such as water), or several fluids completely miscible with each other (e.g. fresh water and salt water). In a multiphase system the pore space is filled with fluids that are immiscible with each other. A typical example is a system consisting of water and oil. Since these two fluids form a distinct boundary between them, each can be considered a separate phase, i.e. the water phase and the oil phase. There may be only a single gaseous phase, however, as all gases are always completely miscible, and the solid matrix of the porous medium can also be treated as a phase called the *solid phase*.

A *component*, also defined in [11], is a part of a phase that is composed of a single chemical species or a collection of species, depending on the physical process that is being modeled. In the case of the fresh water and salt water mixture above, the system consists of a single phase, i.e. water, and two components, i.e. fresh water and salt water.

2.1.2 Continuum Approach

In modeling flow in porous media, it is critical that we consider different length scales as they play an important role in dictating the behavior of the flow. At the smallest scale of about 10^{-9}m (the *molecular scale*), individual molecules of different fluids can be identified, whereas at the *microscopic scale* (about 10^{-3}m), individual soil particles and the pore channels are visible. At largest scale, called the *macroscopic scale*, different types of soils, sands, and rocks, can be differentiated by the difference in their average grain sizes. While it would be ideal to be able to simulate the flow down to the molecular level, such capability is still beyond the reach of current computers and methods. To overcome this difficulty, the fluid is assumed to exist as a continuum on the microscopic scale. Accordingly, quantities such as mass density and velocity are considered continuous functions in space and time. Under this hypothesis, the flow of fluids through the void space of a porous medium can be described by the Navier-Stokes equations with appropriate boundary conditions [29]. Although one can directly solve the Navier-Stokes equations for the velocity of the fluids in a three-dimensional domain obtained from real rock images, it is still a computationally challenging task for complex domains like that shown in figure 2.1[1]. Thus, this approach is only feasible for problems on the microscale (about 10^{-3}m). For applications considered in this work, including groundwater remediation, reservoir simulation, and carbon sequestration, the domain of interest usually ranges from hundreds of meters to kilometers, and direct simulation is not a viable choice. Here, we are interested in simulating the flow on the macroscopic scale and to do so, we need to bridge the gap between the microscopic and macroscopic scales. The idea is to use a representative elementary volume (REV), defined on the microscopic scale, which helps determine the values of physical quantities at a point on the macroscopic scale through an averaging process. In this work, we employ an averaging procedure introduced by [11].

Using the concept of REV, we can define the macroscopic quantities involved

[1] Reprinted from Journal of Petroleum Science and Engineering, Vol 159, N. Zamani, I. Bondino, R. Kaufmann, and A. Skauge, Computation of polymer in-situ rheology using direct numerical simulation, 92-102, Copyright (2017).

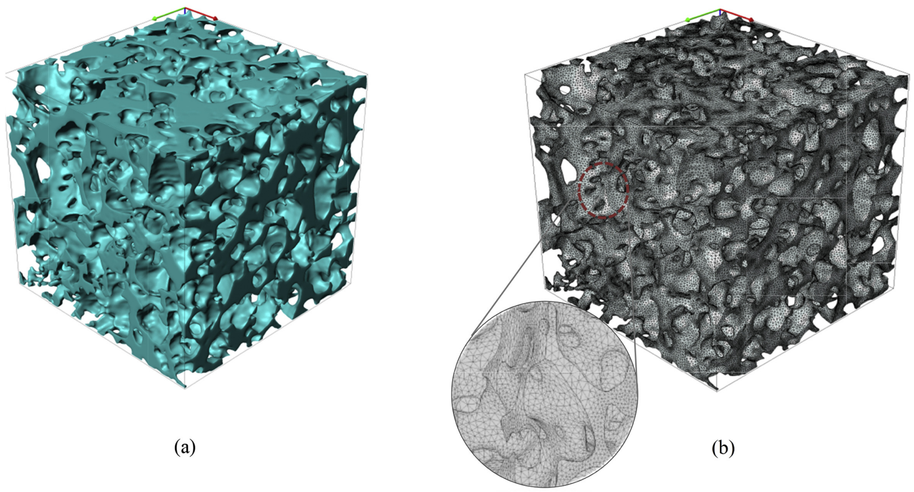


Figure 2.1: (a) μ-CT scan of a sand-pack, (b) generated tetrahedral mesh.

Type of material	Porosity
Consolidated sandstones	0.1-0.3
Uniform spheres with minimal porosity packing	0.26
Uniform spheres with normal packing	0.35
Unconsolidated sands with normal packing	0.39-0.41
Soils with structure	0.45-0.55

Table 2.1: Porosity values for various types of medium

in modeling flow in porous media. In a porous medium, *porosity* ϕ characterizes how much void space there is inside the medium, and it is defined as the ratio between the void space and the total volume of a given REV. By definition, ϕ is between 0 and 1. Table 2.1 (from [36]) shows the porosity values for various media. The void space inside the porous medium is filled with fluids that can exist in one or more phases. The *saturation* of a phase α is defined as the ratio between the volume of that phase over the total volume of the pore space inside a given REV. Thus, by definition, it is immediately clear that the saturation S_α must satisfy the constraint

$$\sum_\alpha S_\alpha = 1, \quad 0 \le S_\alpha \le 1. \tag{2.1}$$

2.1.3 Heterogeneity and Anisotropy

A porous medium is said to be *homogeneous* with respect to a macroscopic (averaged) quantity if that parameter value is constant throughout the domain. Otherwise, if different regions inside the domain can have different parameter values, the medium is called *heterogeneous*. For example, a region that has loose soil packing on top of fine layers of sand is heterogeneous with respect to porosity. In addition to variation with respect to location, macroscopic quantities of a porous medium can also vary with direction. This property is called *anisotropy*.

2.2 General Form of Multiphase Flow Equations

In a system with M phases, the equation for conservation of mass for each phase in a domain $\Omega \in \mathbb{R}^3$ reads

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla \cdot \left(\rho_\alpha \boldsymbol{q}_\alpha\right) = \mathcal{F}_\alpha. \tag{2.2}$$

This equation states that the rate of change of the fluid mass in a control volume $V \in \Omega$ must be equal to the net flow over the surface ∂V plus any source or sink terms within V. Here, ϕ, S_α, ρ_α, $\boldsymbol{q}_\alpha$, and $\mathcal{F}_\alpha$ are the saturation, mass density, Darcy velocity, and forcing term of phase α, respectively.

2.2.1 Constitutive Law

At the microscopic level, the flow takes place through the pore channels with varying velocity. The flow is faster in the regions with smaller pores, while it is slower in those with larger pores. On the macroscopic scale, however, we are only interested in the macroscopic phase velocity (also called Darcy velocity), which is defined to be the average volume of fluid flowing through a unit cross-sectional area per unit time. By applying local averaging techniques (see [96]) or homogenization in [55] to the Navier-Stokes equations, one can derive Darcy's law, which was first discovered by Henry Darcy in 1856. In multiphase flow, an extension of Darcy's law is assumed and the macroscopic phase velocities are expressed as

$$\boldsymbol{q}_\alpha = -\boldsymbol{K}\frac{k_{r\alpha}}{\mu_\alpha}(\nabla P_\alpha - \rho_\alpha \boldsymbol{g}). \tag{2.3}$$

where $\boldsymbol{K}$ is a tensorial quantity specifying the absolute permeability of the porous medium; $k_{r\alpha}$, μ_α, and P_α are the relative permeability, viscosity, and pressure of phase α, respectively; $\boldsymbol{g}$ is the gravity vector pointing in the direction of gravity with the magnitude of gravitational acceleration constant g. Usually, the ratio $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$ is called the mobility of phase α.

Since there is only a fixed void space for the fluids to flow through in a rigid porous medium, the relative permeability terms $k_{r\alpha}$ serve as models to the fact that the flow of one phase hinders the others in a multiphase system. These terms are parametrized by the phase saturation and obey the constraint

$$0 \leq k_{r\alpha}(S_\alpha) \leq 1 \tag{2.4}$$

The phase pressures P_α are related through capillary pressure P_c, which is the jump in the pressure moving from one fluid phase to another. Similar to relative permeability, capillary pressure is parametrized by phase saturation.

$$P_\beta = P_\alpha - P_c(S_\alpha), \quad \forall \beta \neq \alpha. \tag{2.5}$$

Since capillary pressure and relative permeability dictate the nonlinearity of the problem, we will revisit them in separate subsections below.

2.2.2 Relative Permeability and Capillary Pressure

As mentioned above, relative permeability models the fact that the phases obstruct each other in a multiphase system. It is one source of nonlinearity and should be treated carefully. Relative permeability can either be computed by interpolation from a table of measurements for real flow data, or it can be derived from analytical formulation. In this work, the latter approach is taken since the aim is to develop solvers and preconditioners for general multiphase flow. We use two common relations for the relative permeability, the *Brooks-Corey* and *Van Genuchten* models given below.

In the Brooks-Corey model [36], the relative permeabilities for a two-phase system can be expressed in terms of saturation as

$$S_{\alpha e} = \frac{S_\alpha - S_{r\alpha}}{1 - \sum_\beta S_{r\beta}} \tag{2.6}$$

$$kr_w(S_w) = S_{we}^{3+2/\lambda} \tag{2.7}$$

$$kr_n(S_n) = S_{ne}^2\big(1 - (1 - S_{ne})^{1+2/\lambda}\big) \tag{2.8}$$

The relations for the Van Genuchten model [91] read

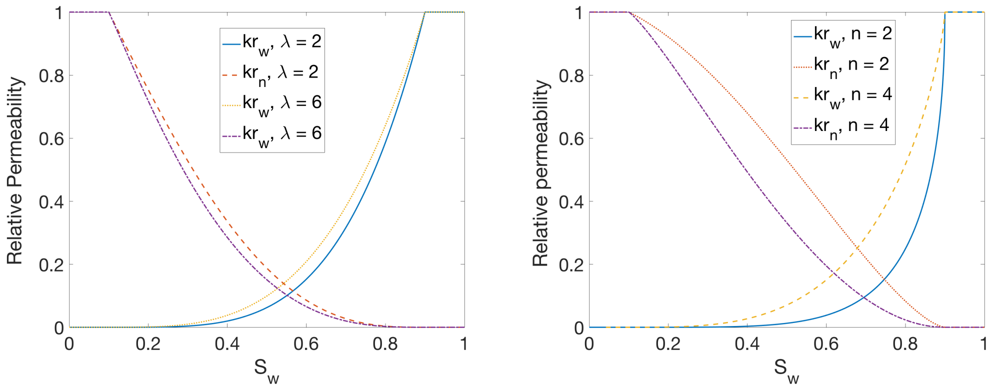
$$k_{rw}(S_w) = \sqrt{S_{we}}\big(1 - \big(1 - S_{we}^{1/m}\big)^m\big)^2, \tag{2.9}$$

$$k_{rn}(S_w) = \sqrt{1 - S_{we}}\big(1 - S_{we}^{1/m}\big)^{2m}. \tag{2.10}$$

In the equations above, the subscripts w, n denote the wetting and non-wetting phases, respectively, and $S_{\alpha e}$ and $S_{\alpha r}$ are the effective and *residual saturations* of phase α, respectively. The values of parameters λ and m depend on the characteristics of the porous media. Particular values for these parameters are listed in the data for the numerical experiments in chapters 3 to 5. Figure 2.2 illustrates the relative permeability curves for different parameters.

The performance of these conventional closed-form models, Brooks-Corey and Van Genuchten, is dependent on the specific shape of the water retention curve for a particular type of soil. The Van Genuchten model is known to more accurately describes S-shaped retention curves characterizing finer-textured soils, whereas the Brooks-Corey model is much better adapted for J-shaped retention curves characterizing relatively coarse-textured soils (see [89]).

The residual saturation of a phase represents the irreducible portion of that phase by pure displacement. For example, as the reservoir is drained, the wetting phase saturation decreases and capillary pressure increases until the the latter becomes so strong that it prevents the wetting phase from escaping the pores. Thus, the capillary pressure curves are extremely steep when the wetting phase saturation approaches the residual saturation (near $S_{we} = 0$ in figure 2.3). It is this large derivative of the capillary pressure function that will require special care in the nu-



(a) Brooks-Corey model.

(b) Van Genuchten model

Figure 2.2: Relative permeability curves for different parameters and residual saturations $S_{wr} = S_{nr} = 0.1$.

merical solution and motivate the development of more robust preconditioners. It is also important to note that even though the phase saturation cannot be reduced below the residual saturation by displacement, it can be reduced by phase transition, such as vaporization or condensation.

Besides relative permeability, another nonlinear effect that strongly affects the flow behavior is capillary pressure. At the microscopic level, there exists a discontinuity at the interface between the wetting and non-wetting phases, and the height of the jump is the capillary pressure, i.e.

$$P_c = P_n - P_w. \tag{2.11}$$

On the macroscopic scale, similar to the relative permeability, the averaged capillary pressure is parametrized by saturation. Again, we use the capillary pressure - saturation relations given by Brooks-Corey and Van Genuchten.

Brooks-Corey capillary pressure curve [36]

$$P_c(S_w) = P_r S_{we}^{-1/\lambda}. \tag{2.12}$$

Van Genuchten capillary pressure curve [91]

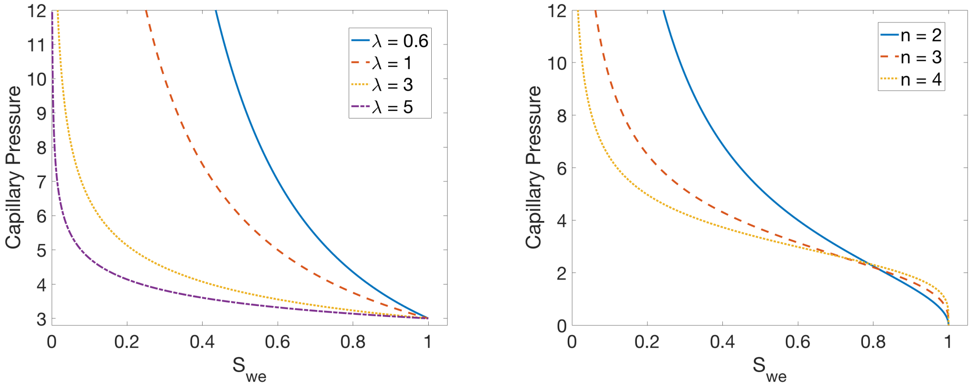
$$P_c(S_w) = P_r(S_{we}^{-1/m} - 1)^{-1/n}, \tag{2.13}$$

$$m = 1 - 1/n. \tag{2.14}$$

The *entry pressure* P_r is the minimum pressure that must be applied for the nonwetting phase to enter the largest pores of the porous medium. In the Van Genuchten model, capillary pressure rises rapidly to the value of P_r for very small change in the saturation near $S_{we} = 1$ (see figure 2.3b), while in the Brooks-Corey model, it is bounded below by P_r (see figure 2.3a). The correct treatment of the entry pressure is especially important since it strongly affects the infiltration rate of the non-wetting phase. An example that shows the difference in the profile of the non-wetting phase for different values of entry pressure is presented in section 5.5.1.

2.3 Incompressible Two-phase Flow

We consider a particular case of immiscible and isothermal two-phase flow through a porous medium. There are numerous instances of such flow in nature. For example, petroleum reservoirs are filled with oil and water. Oil is referred to as the nonwetting phase, and water is the wetting phase. Alternatively, in groundwater management, one may consider a system of contaminated water (wetting phase) that infiltrates a domain saturated with air (nonwetting phase). Due to the incom-



(a) Brooks-Corey model.

(b) Van Genuchten model.

Figure 2.3: Capillary pressure curves for different parameters with residual saturations $S_{wr} = S_{nr} = 0$ and $P_r = 3$.

pressibility assumption, which states that the phase densities ρ_α are constant, the system of conservation equations for multiphase flow in equation (2.2) reduces to

$$\phi\rho_\alpha\frac{\partial S_\alpha}{\partial t} + \rho_\alpha\nabla\cdot\boldsymbol{q}_\alpha = F_\alpha, \quad \alpha = w, n \tag{2.15}$$

The subscripts w, n denote the wetting and nonwetting phases, respectively. Substituting the Darcy velocity in equation (2.3) into equation (2.15) and using the constraint equation (2.1) as well as the relation between phase pressure through capillary pressure in equation (2.5), we have the complete system of equations

$$\phi\rho_w\frac{\partial S_w}{\partial t} - \rho_w\nabla\cdot\big(\boldsymbol{K}\lambda_w(\nabla P_w - \rho_w\boldsymbol{g})\big) = F_w, \tag{2.16}$$

$$\phi\rho_n\frac{\partial S_n}{\partial t} - \rho_n\nabla\cdot\big(\boldsymbol{K}\lambda_n(\nabla P_n - \rho_n\boldsymbol{g})\big) = F_n, \tag{2.17}$$

$$P_w = P_n - P_c \tag{2.18}$$

$$S_w + S_n = 1. \tag{2.19}$$

2.4 Decoupled Model for Incompressible Two-phase Flow

2.4.1 Pressure Equation

In this section, we derive the pressure equation used in simulation of two-phase fluid flow. Dividing both sides of equations (2.16) and (2.17) by the density and summing them together gives

$$\phi\frac{\partial(S_w+S_n)}{\partial t}-\nabla\cdot\big(\lambda_w\boldsymbol{K}(\nabla P_w-\rho_w\boldsymbol{g})+\lambda_n\boldsymbol{K}(\nabla P_n-\rho_n\boldsymbol{g})\big)=\frac{\mathcal{F}_w}{\rho_w}+\frac{\mathcal{F}_n}{\rho_n} \qquad (2.20)$$

Because of the constraint $S_w+S_n=1$, the derivative with respect to time vanishes. Let us introduce the total velocity

$$\boldsymbol{q}=\boldsymbol{q}_w+\boldsymbol{q}_n=-\big(\lambda_w\boldsymbol{K}(\nabla P_w-\rho_w\boldsymbol{g})+\lambda_n\boldsymbol{K}(\nabla P_n-\rho_n\boldsymbol{g})\big) \qquad (2.21)$$

and total source term

$$\mathcal{F}_{tot}=\frac{\mathcal{F}_w}{\rho_w}+\frac{\mathcal{F}_n}{\rho_n}. \qquad (2.22)$$

Equation (2.20) then becomes an elliptic PDE of the form

$$\nabla\cdot\boldsymbol{q}=\mathcal{F}_{tot} \qquad (2.23)$$

Further, let us introduce fractional flow quantity

$$f_\alpha=\frac{\lambda_\alpha}{\lambda}, \qquad \alpha=w,n \qquad (2.24)$$

where $\lambda=\lambda_w+\lambda_n$ is the total mobility. Then, we can rewrite the total velocity as follows

$$\boldsymbol{q}=-\boldsymbol{K}\lambda\big(\nabla P_w+f_n\nabla P_c(S_w)-\boldsymbol{G}\big) \qquad (2.25)$$

In the above equation, we have substituted $P_n = P_w + P_c(S_w)$. The modified gravity term $\boldsymbol{G}$ is defined as

$$\boldsymbol{G} = \frac{\lambda_w \rho_w + \lambda_n \rho_n}{\lambda} \boldsymbol{g}. \tag{2.26}$$

Notice that the formulation for total velocity in equation (2.25) will have the form of Darcy's law in equation (2.3), if we can find a pressure P such that

$$\nabla P = \nabla P_w + f_n \nabla P_c(S_w) \tag{2.27}$$

In this case equation (2.25) turns into

$$\boldsymbol{q} = -\boldsymbol{K} \lambda \big(\nabla P - \boldsymbol{G} \big) \tag{2.28}$$

Following [23], one form of P is

$$P_{=} P_w + \pi_n \tag{2.29}$$

with

$$\pi_n(S_w) = \int_{S_0}^{S_w} f_n(\xi) \frac{\partial P_c}{\partial \xi}(\xi) d\xi + \pi_0. \tag{2.30}$$

One can check that such P satisfying equations (2.29) and (2.30) also obeys equation (2.28). This pressure P is usually referred to as the global pressure.

2.4.2 Saturation Equation

As for the pressure equation, we start by deriving an equation for saturation from one of the mass balance equations. It is common to use the equation for saturation of the wetting phase, but there is no difference for the non-wetting phase,

$$\phi \rho_w \frac{\partial S_w}{\partial t} + \rho_w \nabla \cdot \boldsymbol{q}_w = \boldsymbol{F}_w. \tag{2.31}$$

Here, we want to express the Darcy's phase velocity $\boldsymbol{q}_w$ in terms of the global pressure P and the total velocity $\boldsymbol{q}$. Again, we have the Darcy's velocities for the phases

$$\boldsymbol{q}_w = -\lambda_w \boldsymbol{K}(\nabla P_w - \rho_w \boldsymbol{g}), \tag{2.32}$$

$$\boldsymbol{q}_n = -\lambda_n \boldsymbol{K}(\nabla P_w + \nabla P_c - \rho_n \boldsymbol{g}). \tag{2.33}$$

Dividing both sides by the mobility and taking the difference to eliminate ∇P_w, we obtain

$$\lambda_n \boldsymbol{q}_w - \lambda_w \boldsymbol{q}_n = \boldsymbol{K} \lambda_w \lambda_n \big(\nabla P_c + (\rho_w - \rho_n)\boldsymbol{g}\big). \tag{2.34}$$

Then, substituting $\boldsymbol{q}_n = \boldsymbol{q} - \boldsymbol{q}_w$ and simplifying gives

$$\boldsymbol{q}_w = f_w \boldsymbol{q} + \lambda_n f_w \boldsymbol{K} \big(\nabla P_c + (\rho_w - \rho_n)\boldsymbol{g}\big). \tag{2.35}$$

2.4.3 Complete System of Equations

Putting all the equations together gives the following system of equations for incompressible 2-phase flow

$$-\nabla \cdot \big(\boldsymbol{K}\lambda(\nabla P - \boldsymbol{G})\big) = \digamma_{tot}, \tag{2.36}$$

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot \boldsymbol{q}_w = \digamma_w / \rho_w, \tag{2.37}$$

$$\boldsymbol{q}_w = f_w \boldsymbol{q} + \lambda_n f_w \boldsymbol{K} \big(\nabla P_c + (\rho_w - \rho_n)\boldsymbol{g}\big), \tag{2.38}$$

$$\lambda = \lambda_w + \lambda_n, \tag{2.39}$$

$$\lambda_w = k_{rw}/\mu_w, \qquad \lambda_n = k_{rn}/\mu_n. \tag{2.40}$$

For the pressure equation, we need to specify either the pressure on the boundary, which leads to a Dirichlet boundary condition

$$P = P_d \quad \text{on} \quad \partial\Omega_D,$$

or the fluxes on the faces, which results in a Neumann boundary condition

$$\boldsymbol{q} \cdot \boldsymbol{n} = \boldsymbol{G}_n \quad \text{on} \quad \partial\Omega_N.$$

Similarly for the saturation equation, we have the following boundary conditions

- Dirichlet boundary condition, in which the saturation on the boundary is given.

$$S = S_d \quad \text{on} \quad \partial\Omega_D.$$

- Neumann boundary condition, in which the fluxes on the boundary faces are given.

$$\boldsymbol{q}_w \cdot \boldsymbol{n} = \boldsymbol{G}_n \quad \text{on} \quad \partial\Omega_N.$$

2.5 Compositional Multiphase Flow

In the multiphase flow models presented in previous sections, we have not taken into account the fact that in reality, the phases rarely consist of a single component. On the contrary, each phase usually contains many different species of fluids such as water, hydrogen, carbon dioxide, etc. To include these components, more detailed models are needed. Before we begin to formulate these models, it is convenient to introduce some fractional quantities that are useful in analyzing the composition of the flow.

Component volume fraction

$$C_\alpha^k = \frac{\text{Volume of component } k \text{ in phase } \alpha}{\text{Volume of phase } \alpha}. \tag{2.41}$$

Mass fraction of a component

$$X_\alpha^k = \frac{\text{Mass of component } k \text{ in phase } \alpha}{\text{Mass of phase } \alpha}. \tag{2.42}$$

Intrinsic mass density of a component

$$\rho_\alpha^k = \frac{\text{Mass of component } k \text{ in phase } \alpha}{\text{Volume of component } k \text{ in phase } \alpha}. \tag{2.43}$$

These fractional quantities are related to each other by the equation

$$\rho_\alpha X_\alpha^k = \rho_\alpha^k C_\alpha^k. \tag{2.44}$$

It is also obvious that we have the following constraints

$$\sum_{k=1}^{N} X_\alpha^k = 1, \quad \sum_{k=1}^{N} C_\alpha^k = 1. \tag{2.45}$$

Then, the terms in the mass balance equations for an M-phase and N-component system can be rewritten as

$$\begin{aligned}
&\frac{\partial \xi^k}{\partial t} + \nabla \cdot \psi^k = \boldsymbol{F}^k, \\
&\xi^k = \phi \sum_{\alpha=1}^{M} \rho_\alpha X_\alpha^k S_\alpha, \\
&\psi^k = \sum_{\alpha=1}^{M} \big(\rho_\alpha X_\alpha^k \boldsymbol{q}_\alpha + D_\alpha^k \rho_\alpha \nabla X_\alpha^k \big),
\end{aligned}$$

where ξ^k, ψ^k are the storage and flux terms of component k, respectively, and D_α^k is the diffusion tensor of component k in phase α. Similar to the incompressible case, the velocity $\boldsymbol{q}_\alpha$ is assumed to satisfy an extension of Darcy's law for multiphase flow, i.e.

$$\boldsymbol{q}_\alpha = \boldsymbol{K}\lambda_\alpha \nabla(P_\alpha - \rho_\alpha \boldsymbol{g}), \quad \alpha = 1, ..., M \tag{2.46}$$

Equation (2.46) states that the total mass of a component k across all the phases must be conserved, and the change in the mass of a component is attributed to the flux through the boundary of a control volume and the mass exchange between the phases due to diffusion.

Chapter 3: Block Factorization Preconditioner for Immiscible Two-phase Flow

3.1 Introduction

In this chapter, we focus on the iterative solution of linear systems arising in a fully implicit cell-centered finite volume discretization of single component isothermal, incompressible two-phase flow model with capillary pressure. This fully implicit time-stepping scheme is among the most robust for simulation of subsurface flow. Moreover, it can serve as a basis for modeling more complex processes in which the physical quantities are tightly coupled. This additional complexity could include adding more components, miscibility between components, thermal effects, and phase transitions. The results included in this chapter also appear in [21].

The fully implicit discretization gives rise to a nonlinear system of equations at each time step. We employ a variant of Newton's method with an exact Jacobian of the discretized equations to solve this system. For the linear system, we use a preconditioned GMRES method [84]. There is a vast literature on different approaches to precondition the Jacobian system. A very popular approach is to use ILU for constructing the preconditioner. Though popular for their general applicability, ILU-based preconditioners are neither effective nor scalable in many cases. Another approach is to consider decoupled preconditioners for the coupled system [12]. This methodology is based on a direct solution of the decoupled pressure system, followed by an iterative solution using ILU for the global system. This formulation was refined in [93], where it was proposed to solve the pressure system iteratively, giving rise to the decoupled IMPES preconditioner. The effect of the decoupling is to weaken the coupling between pressure and saturation. Thus, it is often used as a preprocessing step to produce a modified Jacobian system, for which new preconditioners can be developed [33,87]. Another approach to break up the coupled problem into a sequence of simpler problems include operator splitting techniques, developed in [40,41,70]. With recent development of algebraic multigrid (AMG) algorithms, the pressure block can be solved efficiently using AMG, resulting in the constrained pressure residual multigrid (CPR-AMG) approach. In recent developments, AMG has also been applied to solve the coupled system with some success [33, 87], although developing a general AMG algorithm for these types of problems remains a topic of ongoing research [94]. Since the Jacobian matrix has a block structure, one can also consider a block LU decomposition with an approximate Schur complement, which has been successfully applied to other models of fluid dynamics [68,97]. Besides AMG-based methods, geometric multigrid has also been applied successfully to solve these types of problems [9,10]. However, our focus in this study is on AMG variants because of their general applicability.

In this work, we develop a new block preconditioner designed to respect the coupling inherent in models of multiphase flow, and we report our experience with the performance and scalability of four different preconditioning strategies: (1) a two-stage CPR-AMG method with correction for the pressure block, also known as the combinative two-stage approach, (2) a CPR-AMG with corrections for both the pressure and saturation blocks, known as the two-stage additive approach, (3) the block factorization (BF) preconditioner, and (4) a direct AMG preconditioner for the global system. An outline of the chapter is as follows. In section 3.2, we present the mathematical formulation for two-phase flow in porous media and discretization schemes. In section 3.3, we describe the new solution algorithms that we have developed for the linearized system. Numerical results for the algorithms are presented in section 3.4. We conclude with some remarks and discussion of future research directions in section 3.5.

3.2 Problem Statement

As shown in section 2.4, one can use a decoupling approach for equations (2.16) to (2.19) and derive separate equations for pressure and saturation from system. The pressure equation is elliptic due to incompressibility; the saturation equation is of advection-diffusion type. Depending on the applications and capillary pressure models, the saturation equation can be diffusion-dominated, advection-dominated, or even purely hyperbolic (in the absence of capillary pressure).

The pressure equation is solved implicitly, and depending on the time discretization strategies applied to the saturation equation, several methods have been developed. In the case where the saturation equation is discretized using an explicit method (e.g., forward Euler), it is referred to as IMPES (implicit pressure explicit saturation) [7]; for an implicit time discretization of the saturation equation, the method is known as the sequential approach, which was first applied to the black-oil model in [95].

The appeal of these methods lies in the sequential decoupling between pressure and saturation variables. Each equation can be solved separately. In addition, knowing the features of each equation (for example, direction of the flow or relative sizes of diffusion and advection terms) facilitates the design of efficient preconditioners, which is critical to achieving high performance. Both of these methods have been successfully applied to many problems where the fully implicit method is difficult to implement or shown to be too costly. However, the solution obtained from these approaches may lose accuracy due to poor convergence behavior if pressure and saturation are strongly dependent, or if capillary pressure changes very quickly. The lack of accuracy of these methods can be even more pronounced if more complex processes such as miscibility, thermal, and phase transitions are included in the model. For a more complete summary of the advantages and disadvantages of these approaches, we refer to [61].

In this chapter, we take a different point of view and explore methods for solving the fully coupled system stated in equations (2.16) to (2.19) which avoids the limitations of the decoupling approach mentioned above. Substitution of equation (2.18) into equation (2.17) and using the constraint equation (2.19) yields a system of two equations and two unknowns. Using one popular choice of primary variables, the pressure in the wetting phase and saturation in the nonwetting phase,

$\boldsymbol{u} = (P_w, S_n)$ [98], we obtain

$$-\frac{\partial(\phi\rho_w S_n)}{\partial t} - \nabla\cdot\Big(\rho_w\frac{k_{rw}(S_w)}{\mu_w}\boldsymbol{K}(\nabla P_w - \rho_w\boldsymbol{g})\Big) = \digamma_w, \tag{3.1}$$

$$\frac{\partial(\phi\rho_n S_n)}{\partial t} - \nabla\cdot\Big(\rho_n\frac{k_{rn}(S_n)}{\mu_n}\boldsymbol{K}(\nabla(P_w + P_c(S_n)) - \rho_n\boldsymbol{g})\Big) = \digamma_n. \tag{3.2}$$

This formulation also has the advantage that extending it to the case of compressible flow and multi-component flow is quite straightforward. See also [16, 59] for use of this model. In the fully coupled approach, we consider solving equations (3.1) and (3.2) together as a big system fully implicitly. We use a cell-centered finite volume method for spatial discretization and the backward Euler method for time discretization, similar to an approach defined in [38]. This will serve as a base model for adding more complexity in the future.

The finite volume method described below is known for its mass conservation property. In addition, it can deal with the case of discontinuous permeability coefficients, and it is relatively straightforward to implement. Under appropriate assumptions, this method also falls into the mixed finite element framework [69,81]. For simplicity, we consider a uniform partitioning of the domain $\boldsymbol{\Omega}$ into equal sized cells C_i, i.e., $\boldsymbol{\Omega} = \bigcup\limits_{i=1} C_i$. Let γ_{ij} denote the area of the face between cells C_i and C_j. For each cell C_i, integration of the mass conservation equations and the divergence theorem gives

$$\frac{\partial}{\partial t}\int_{C_i}\xi_\alpha + \sum_{j\in\eta_i}\int_{\gamma_{ij}}\psi_\alpha\cdot\boldsymbol{n} = \int_{C_i}\digamma_\alpha, \tag{3.3}$$

where the *storage* $\xi_\alpha = \phi\rho_\alpha S_\alpha$ and the *flux* $\psi_\alpha = \rho_\alpha\boldsymbol{q}_\alpha$ terms are approximated using the mid-point rule which is second-order accurate:

$$\bar{\xi}_\alpha = \frac{1}{V_{C_i}}\int_{C_i}\xi_\alpha, \quad \bar{\mathcal{F}}_\alpha = \frac{1}{V_{C_i}}\int_{C_i}\mathcal{F}_\alpha. \tag{3.4}$$

The surface integrals are discretized using two-point flux-approximation (TPFA); dropping the phase subscript, this gives

$$\int_{\gamma_{ij}} \psi \cdot \boldsymbol{n} = -\gamma_{ij}\Big(\rho\frac{k_r}{\mu}\boldsymbol{K}\Big)_{ij+1/2}\big(\omega_i - \omega_j\big)\,, \tag{3.5}$$

$$\omega_i = \frac{P_i - \rho_{ij+1/2}\,\boldsymbol{g}_i}{\Delta x_{ij+1/2}}\,. \tag{3.6}$$

The quantities with subscript i, j are defined at the centers of cells i, j, respectively, and the subscript $ij + 1/2$ signifies an appropriate averaging of properties at the interface between cells i and j. The coefficients $(\rho k_r/\mu)_{ij+1/2}$ are approximated by upwinding based on the direction of the velocity field, i.e.,

$$\Big(\rho\frac{k_r}{\mu}\Big)_{ij+1/2} = \begin{cases} \Big(\rho\dfrac{k_r}{\mu}\Big)_i\,, & \text{if } \boldsymbol{v}\cdot\boldsymbol{n} > 0 \\ \Big(\rho\dfrac{k_r}{\mu}\Big)_j\,, & \text{otherwise} \end{cases} \tag{3.7}$$

and the absolute permeability tensor on the faces is computed using harmonic averaging,

$$\boldsymbol{K}_{ij+1/2} = (\Delta x_i + \Delta x_j)\Big(\frac{\boldsymbol{K}_i\boldsymbol{K}_j}{\Delta x_i\boldsymbol{K}_j + \Delta x_j\boldsymbol{K}_i}\Big). \tag{3.8}$$

The TPFA finite volume scheme presented here is convergent only if the mesh is K-orthogonal, i.e. each grid cell is a parallelepiped and

$$\boldsymbol{n}_{ij}\boldsymbol{K}\boldsymbol{n}_{ik} = 0, \quad \forall C_i \in \Omega, \quad \boldsymbol{n}_{ij} \neq \pm\boldsymbol{n}_{ik}, \tag{3.9}$$

where $\boldsymbol{n}_{ij}$ and $\boldsymbol{n}_{ik}$ are the normal vectors from cell C_i into neighboring cells C_j and C_k, respectively. Orthogonal grid is a case of K-orthogonal grid in which the

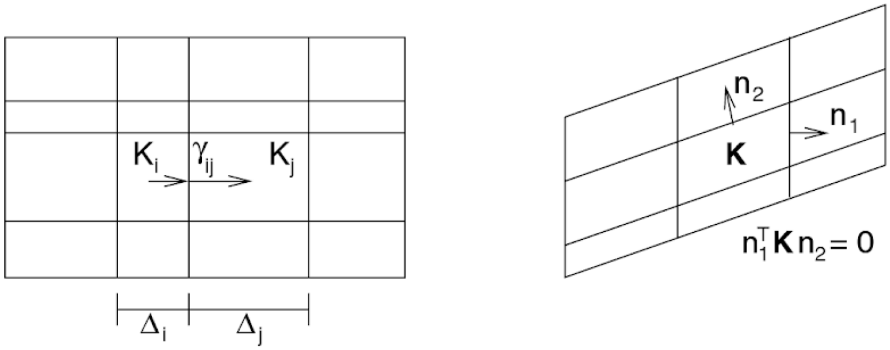


Figure 3.1: (left) simple orthogonal grid and (right) general K-orthogonal grid. Images obtained from [44].

permeability tensor in equation (3.9) is diagonal, as opposed to a full tensor. The difference between simple orthogonal grids and k-orthogonal grids is illustrated in figure 3.1. We note that for simplicity, structured Cartesian grids are employed for the test cases in section 3.4, but the method and our implementation can accommodate unstructured K-orthogonal grids (see figure 3.1). The backward Euler method is used for discretization of the time domain. This method is unconditionally stable and robust with respect to adaptive time-stepping schemes because it is fully implicit and there is no CFL stability restriction of the time step. Application of the backward Euler method gives a fully discrete system of nonlinear equations,

$$(\bar{\xi})_i^{n+1} - (\bar{\xi})_i^{n} = -\frac{\triangle t}{V_{C_i}} \sum_{j \in \eta_i} \gamma_{ij} \left(\rho \frac{k_r}{\mu} \boldsymbol{K} \right)^{n+1}_{ij+1/2} \left(\omega_i^{n+1} - \omega_j^{n+1} \right) - F^{n+1}. \qquad (3.10)$$

3.3 Solution Algorithms

The system of nonlinear equations (3.10) which must be solved at each time step, can be written generically as $F(u) = 0$ where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. We solve the system using Newton's method, which requires solution of a linear system at each iteration k:

$$\left.\frac{\partial F}{\partial u}\right|_{u=u_k}(u_{k+1}-u_k) = -F(u_k). \tag{3.11}$$

In our case, the solution vector u consists of all the pressure and saturation unknowns at all the cell centers. The Jacobian system resulting from the derivative $\partial F/\partial u$ is often very difficult to solve using iterative methods, and preconditioning is critical for rapid convergence of Krylov subspace methods such as GMRES. Next, we discuss the linear system arising from Newton's method and give a detailed description of the solution algorithms we will use to solve this system.

3.3.1 Linear System

For the set of primary variables $u = (P_w, S_n)$, assuming that each physical variable is ordered lexicographhpically from left to right and bottom to top in the domain, then each nonlinear Newton iteration entails the solution of a discrete version of a block linear system of the form

$$\begin{pmatrix} -\nabla\cdot(\lambda_w \boldsymbol{K}\nabla) & -\dfrac{\phi}{\partial t} - \nabla\cdot(\boldsymbol{q}_w) \\ -\nabla\cdot(\lambda_n \boldsymbol{K}\nabla) & \dfrac{\phi}{\partial t} + \boldsymbol{q}_n\cdot\nabla + \nabla\cdot(\lambda_n P_c' \boldsymbol{K}\nabla) \end{pmatrix} \begin{pmatrix} \delta P_w \\ \delta S_n \end{pmatrix} = - \begin{pmatrix} F_w \\ F_n \end{pmatrix}, \tag{3.12}$$

in which

$$\boldsymbol{q}_w = -\lambda_w' \boldsymbol{K}\nabla\tilde{P}_w, \tag{3.13}$$

$$\boldsymbol{q}_n = -\lambda_n' \boldsymbol{K}\nabla\tilde{P}_n + \lambda_n \boldsymbol{K}\nabla(P_c')\ . \tag{3.14}$$

All the coefficients in equation (3.12) are evaluated at the linearization point $\tilde{P}_w, \tilde{S}_n$. In a more concise form, the Jacobian matrix of the system has 2×2 block structure

$$J = \begin{pmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{pmatrix}, \tag{3.15}$$

and the linear system is $Jc = q$. The characteristics of the matrix have been discussed in numerous papers [10, 38, 57, 87]. We summarize important characteristics of the operators here:

- J is nonsymmetric and indefinite
- The block A_{pp} has the structure of a discrete purely elliptic problem for pressure.
- The coupling block A_{ps} has the structure of a discrete first-order hyperbolic problem in the non-wetting phase saturation.
- The coupling block A_{sp} has the structure of a discrete advection-free parabolic problem in the wetting phase pressure.
- The block A_{ss} has the structure of a discrete parabolic (advection-diffusion) problem for saturation when capillary pressure is a non-constant function of the saturation. When capillary pressure is zero or a constant, there is no diffusion term and the block has the form of a hyperbolic problem.
- Under mild conditions, i.e. modest time-step size, the blocks A_{pp}, A_{ps}, A_{ss} are diagonally dominant.

In this chapter, we present some numerical results that show how different models of capillary pressure affect the algebraic properties of the (2,2)-block A_{ss} in particular and the global system in general, which consequently determines the success of AMG solution algorithms. Our emphasis is on the development and use of preconditioning operators denoted $M \approx J$, for the purpose of solving preconditioned systems

$$JM^{-1}\hat{c} = q, \quad c = M^{-1}\hat{c}. \tag{3.16}$$

3.3.2 Two-stage Preconditioning with AMG

In the context of multiphase flow modeling, two-stage preconditioners first appeared in [93], and since then they have been widely used in reservoir simulation [57]. Following [38], we refer to this method as the constrained pressure residual (CPR) approach. There are many variants of two-stage preconditioners. We discuss two algorithms here: the two-stage combinative preconditioner - CPR-AMG(1), and the two-stage additive preconditioner - CPR-AMG(2) [6].

Algorithm 1 Two-stage Combinative - CPR-AMG(1)

```
1. At each iteration $k$ let the residual be $r_k = f_k - Au_k$.
2. Solve $\delta u_{k+1/2} = P_1^{-1} r_k$, compute intermediate solution $u_{k+1/2} = u_k + \delta u_{k+1/2}$.
3. Update the residual $r_{k+1/2} = r_k - A\delta u_{k+1/2}$.
4. Solve for the pressure correction $A_{pp}\delta_p = R_p r_{k+1/2}$.
5. Update the solution $u_{k+1} = u_{k+1/2} + R_p^T \delta_p$.
```

Algorithm 2 Two-stage Additive - CPR-AMG(2)

```
1. At each iteration $k$ let the residual be $r_k = f_k - Au_k$.
2. Solve $\delta u_{k+1/2} = P_1^{-1} r_k$, compute intermediate solution $u_{k+1/2} = u_k + \delta u_{k+1/2}$.
3. Update the residual $r_{k+1/2} = r_k - A\delta u_{k+1/2}$.
4. Solve for the pressure correction $A_{pp}\delta_p = R_p r_{k+1/2}$.
5. Solve for the saturation correction $A_{ss}\delta_s = R_s r_{k+1/2}$.
6. Update the solution $u_{k+1} = u_{k+1/2} + R_p^T \delta_p + R_s^T \delta_s$.
```

The matrices R_p, R_s denote the restriction of the global unknown vector to the spaces associated with pressure and saturation respectively. That is, $R_p \in \mathbb{R}^{n\times 2n}$ and for $u = \begin{pmatrix} p \\ s \end{pmatrix}$

$$R_p u = p, \quad R_p^T u = \begin{pmatrix} p \\ 0 \end{pmatrix}; \qquad R_s u = s \quad R_s^T u = \begin{pmatrix} 0 \\ s \end{pmatrix}. \tag{3.17}$$

Then, in matrix form, the action of the two-stage preconditioners can be expressed as

$$\delta u = M_{comb}^{-1} r = (I - R_p^T A_{pp}^{-1} R_p (A - P_1)) P_1^{-1} r, \tag{3.18}$$

$$\delta u = M_{add}^{-1} r = (I - (R_p^T A_{pp}^{-1} R_p + R_s^T A_{ss}^{-1} R_s)(A - P_1)) P_1^{-1} r. \tag{3.19}$$

The preconditioner P_1 in step 2 of both algorithms is taken to be the ILU(0) factorization of A, i.e., the incomplete factorization with no fill applied to the global matrix. For the correction solve, we use AMG with one V-cycle iteration. The combinative approach with AMG was presented in [58]. However, this method does not work well in the presence of fast changing capillary pressure. We confirm this observation in the next section. To deal with fast changing capillary pressure, we employ an additive CPR-AMG approach, which involves one more AMG solve for the correction of the saturation block. The intuition is that when the absolute value of the derivative of capillary pressure $|dP_c/dS_w|$ is large, the block A_{ss} becomes diffusion dominated, and AMG can handle it efficiently.

3.3.3 Block Factorization Preconditioners

Consider the following decomposition of the Jacobian,

$$J = \begin{pmatrix} A_{pp} & A_{ps} \\ A_{sp} & A_{ss} \end{pmatrix} = \begin{pmatrix} I & A_{ps}A_{ss}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} S & 0 \\ 0 & A_{ss} \end{pmatrix} \begin{pmatrix} I & 0 \\ A_{ss}^{-1}A_{sp} & I \end{pmatrix},$$

where S is the Schur complement

$$S = A_{pp} - A_{ps}A_{ss}^{-1}A_{sp}. \tag{3.20}$$

We could choose

$$M = \begin{pmatrix} I & A_{ps}A_{ss}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} S & 0 \\ 0 & A_{ss} \end{pmatrix} = \begin{pmatrix} S & A_{ps} \\ 0 & A_{ss} \end{pmatrix} \tag{3.21}$$

as an upper-triangular block preconditioner; this incorporates the effects of the coupling block A_{ps}. This block is important as it contains the time derivative and gravity terms (see equation (3.12)). If the time step is small, then the coefficients on the diagonal of this block become large and it is important that this term be included in the preconditioner. We use an approximation of the Schur complement in which A_{ss} is replaced by its diagonal values:

$$\tilde{S} = A_{pp} - A_{ps}\,\mathrm{diag}\,(A_{ss})^{-1} A_{sp}. \tag{3.22}$$

The purpose of this is to keep the Schur complement sparse so that the action of its inverse can be applied efficiently. This idea is the basis of the SIMPLE method used in other models of fluid dynamics [68]. A similar approach has also been applied to problems in single phase flow coupled with geomechanics in [97].

Algorithm 3 Block factorization preconditioner

```
1. At each iteration $k$ let the residual be $r_k = f_k - Au_k$.
2. Solve for the saturation $A_{ss}s_{k+1} = R_s r_k$ using AMG.
3. Compute the residual for pressure $r = R_p r_k - A_{ps}s_{k+1}$.
4. Solve for the pressure $\tilde{S}p_{k+1} = r$ using AMG.
```

An important advantage of this algorithm is that it does not rely on an ILU factorization. In matrix form,

$$M_{bf}^{-1} = \begin{pmatrix} \tilde{S}^{-1} & -\tilde{S}^{-1}A_{ps}A_{ss}^{-1} \\ 0 & A_{ss}^{-1} \end{pmatrix}. \tag{3.23}$$

3.3.4 Algebraic Multigrid

Multigrid is a highly efficient and scalable method available for solving large sparse linear systems [88,100]. Geometric multigrid uses a hierarchy of nested grids, whose construction depends on the geometry of the problem and *a priori* knowledge of the grids. AMG methods such as those developed in [86] have the advantage of not requiring an explicit hierarchy of nested grids. AMG constructs coarse grids based on the matrix values only, which makes it suitable for solving a wide range of problems on complicated domains and unstructured grids. Despite its successful application to scalar problems, using AMG for coupled systems is still relatively limited. Some attempts to use AMG to solve fully coupled systems encountered in modeling multiphase flow for reservoir simulation include [33, 87]. In this work, BoomerAMG [52] from Hypre [48, 49] is used as a black-box AMG preconditioner for performance comparison in section 3.4. We note that in our implementation of the coupled system, the Jacobian matrix passed to BoomerAMG is ordered by grid points, i.e.

$$J = \begin{pmatrix} A_{11} & \dots & A_{1N} \\ \vdots & \ddots & \dots \\ A_{N1} & \dots & A_{NN} \end{pmatrix}, \tag{3.24}$$

in which N is the number of grid points, and A_{ij} are 2×2 matrices representing the couplings between pressure and saturation at points i and j, and we use this ordering for tests of AMG. This is called the “point” method in [87].

3.4 Numerical Results

To verify the correctness of the fully coupled approach, we implement the sequential method for the decoupled model presented in section 2.4 and treat the solutions produced by this method as reference solutions for the test problems. In the sequential method, we first solve the pressure equation (equation (2.36)) for pressure, and then use this pressure field to advance the saturation using equation (2.37). To verify the pressure solver (for the decoupled model), we examine the following test problem

$$
\begin{aligned}
-\nabla \cdot (\boldsymbol{K} \nabla P) &= F && \text{on } \Omega = [0,1] \times [0,1] \\
P &= P_D && \text{on } \partial\Omega.
\end{aligned}
$$

with the parameters

$$
\begin{aligned}
\boldsymbol{K} &= 1, P_D = 0, \\
F &= 5\pi^2 \sin(\pi x) \sin(2\pi y).
\end{aligned}
$$

The exact solution reads

$$
P_{exact} = \sin(\pi x) \sin(2\pi y).
$$

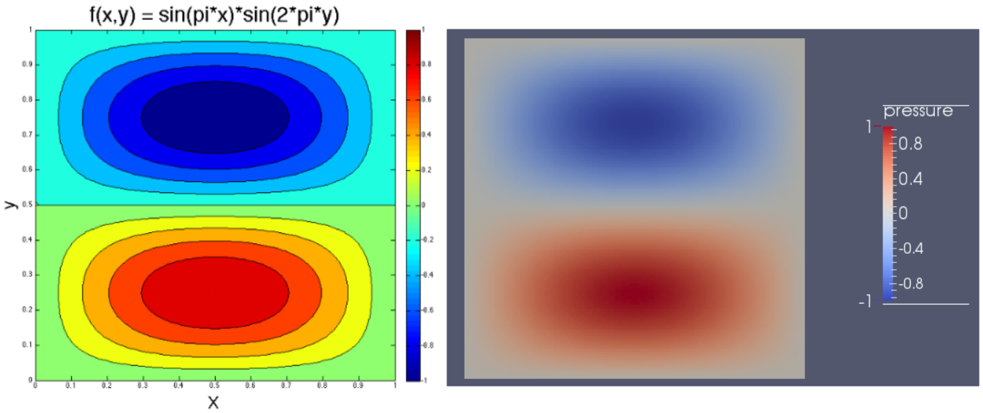


Figure 3.2: Analytic solution (left) and numeric solution (right) for test problem

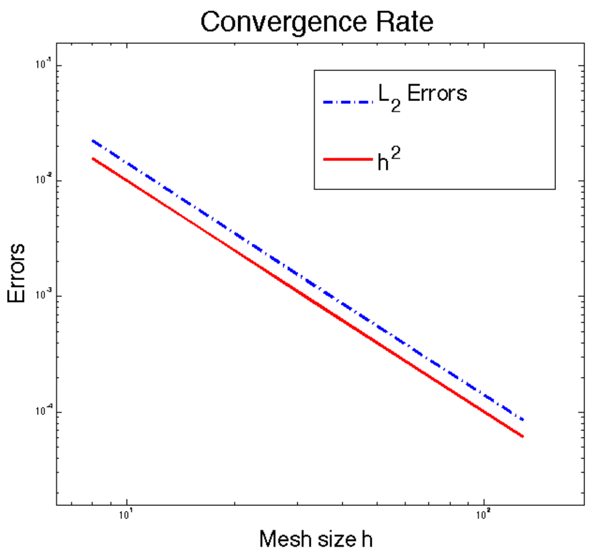


Figure 3.3: Convergence rate for test problem, with mesh size $h = 8, 16, 32, 64, 128$.

As can be seen in figure 3.2, the analytic and numerical solutions are consistent with each other, which indicates that our implementation of the pressure solver is correct. This is further confirmed by a convergence test. Figure 3.3 shows that the finite volume scheme with TPFA achieves the second-order accuracy as predicted by theory, i.e. the L_2 errors $e = ||\hat{p} - p_{exact}||$ is proportional to h^2 (h is the mesh size).

For the saturation equation, we use the one-dimensional Buckley-Leverett equation given by

$$\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x}\Big(\frac{Q}{\phi A} f(S_w)\Big) = 0,$$

where S_w is the saturation of the wetting phase, Q is the flux, A is the cross section area, ϕ is the porosity, and f is the fractional flow. The Buckley-Leverett equation is a nonlinear hyperbolic equation with non-convex flux

$$f(S_w) = \frac{S_w^2}{S_w^2 + (1 - S_w)^2}. \tag{3.25}$$

For simplicity, we use unit flux $Q = 1$, unit cross-sectional area $A = 1$, and unit porosity $\phi = 1$. The initial condition is as follows

$$S_w(t = 0) = \begin{cases} 1 \text{ for } x \leq 0.2 \\ \\ 0 \text{ for } x > 0.2 \end{cases}. \tag{3.26}$$

For boundary conditions, we have Dirichlet boundary conditions $S_w(x = 0) = 1$ and $S_w(x = 1) = 0$. The analytic and numerical solutions to this problem are plotted in figure 3.4.

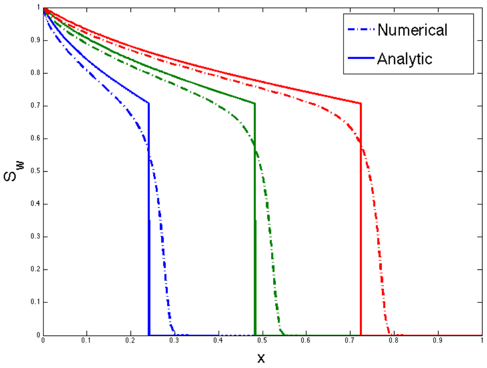


Figure 3.4: Solutions of the Buckley-Leverett equation for different time steps using first order upwind scheme: $t = 0.2$ (blue), $t = 0.4$ (green), and $t = 0.6$ (red).

Similar to the pressure equation, we use finite volume with upwind stabilization for spatial discretization. In this approach, the values of the flux $f(S)$ on the faces is evaluated using the cell-valued saturation based on the direction of the velocity (see equation (3.7)). From figure 3.4, it is clear that the backward Euler scheme

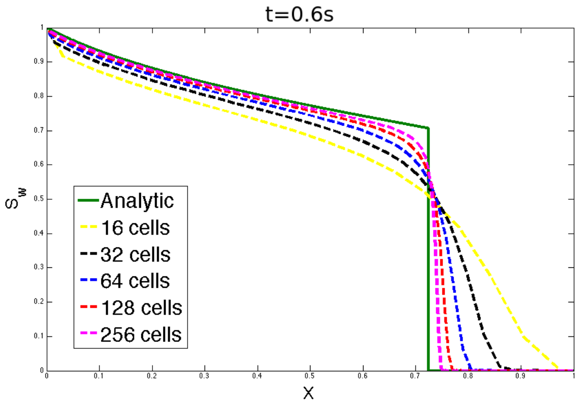


Figure 3.5: Numerical solutions with different mesh sizes at time $t = 0.6s$

Elements	Time Steps	L_1 Error	Rate	L_2 Error	Rate
16	32	0.124		0.171	
32	64	0.0779	0.67	0.135	0.35
64	128	0.0475	0.71	0.105	0.35
128	256	0.0283	0.75	0.082	0.36
256	512	0.0165	0.78	0.063	0.38

Table 3.1: Convergence rate for L_1 and L_2 error for the Buckley-Leverett equation.

with upwinding suffers from numerical diffusion, which smooths out the sharp front observed in the analytic solution. However, as can be seen in figure 3.5, the numerical solution approaches the analytic solution as the mesh resolution increases. We compute the L_1 and L_2 errors and the convergence rate and report them in table 3.1. Note that we exclude a fixed layer near the shock when computing the errors. In this region, the convergence rate does not follow the same order as in the rest of the domain [45]. The convergence rates in table 3.1 are comparable to those listed in [9]. While the Buckley-Leverett is a scalar hyperbolic PDE, it can be used as a test problem for the fully coupled incompressible two-phase system in equations (3.1) and (3.2) if the pressure gradient is constant. For this test problem, we note that the fully coupled approach produces nearly identical profiles for the saturation as presented in figure 3.5, and it also achieves similar convergence rates to those in table 3.1, which indicates that our implementation of the fully coupled approach is correct.

In this section, we perform numerical experiments for the four aforementioned preconditioners: (1) the combinative two-stage approach CPR-AMG(1) (Algorithm 1), (2) the two-stage additive approach CPR-AMG(2) (algorithm 2), (3) the block factorization (BF) preconditioner in equation (3.23), and (4) a direct AMG preconditioner for the global system. All of them are implemented in Amanzi, a parallel open-source multi-physics C++ code developed as a part of the Advanced Simulation Capability for Environmental Management (ASCEM) project [5]. Amanzi uses an object-oriented design that mirrors the hierarchy of processes being modeled to provide both run-time flexibility and significant extensibility in adding new models (process kernels). It also provides parallel distributed memory infrastructure for unstructured and structured AMR meshes, advanced discretization techniques such as the mimetic finite difference (MFD) method, and linear and nonlinear solvers. Where possible, Amanzi leverages existing open-source libraries such as Trilinos [53] and Hypre, rather than duplicating capability and effort. Although Amanzi was first designed for simulation of subsurface flow and reactive transport, its modular framework and concept of process kernels [35] allow new physics to be added relatively easily for other applications. The two-phase flow simulator that we developed for this work is one such example. Amanzi works on a variety of platforms, from laptops to supercomputers. It also leverages several popular packages for mesh infrastructure and solvers through a unified input file. Here, all of our experiments use a classical AMG solver through BoomerAMG in Hypre. The ILU(0) method is from Euclid, also a part of Hypre. ILU(0) is used sequentially for the two-dimensional examples, and parallel ILU(0) (also from Euclid) is used for the three-dimensional cases. GMRES is provided within Amanzi. The test cases are run on Edison, a Cray system at the National Energy Research Scientific Computing Center (NERSC)[1], with a peak performance of 2.57 petaflops/sec, 133,824 compute cores, and 357 terabytes of memory. We run the two-dimensional test cases in serial, and the three-dimensional SPE10 problem (see section 3.4.4) with 256 cores. Amanzi and other libraries are compiled with OpenMPI 1.6.5 and gcc-4.9.2. The total time is measured in seconds.

This section has three parts. In the first part, we show the results for a two-dimensional oil-water model problem. Although the problem is small, it is difficult to solve due to the heterogeneity of the permeability field. In the second part, we report the results for a three-dimensional example. In the last part, we examine the scalability of the three preconditioning strategies. Unless specified otherwise, we use the benchmark problem SPE10 [28] for permeability data and porosity.

[1] A DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

3.4.1 Two-dimensional oil-water problem

The domain is a rectangle with dimensions 762×15.24 meters. The mesh is 100×20, which means that the problem is truly two-dimensional in the xz plane. We inject pure water into the domain through the boundary at the lower left corner, and oil and water exit the domain through the top right corner. These correspond to the $S_w = 1.0$, $\lambda_w \nabla P_w \cdot \boldsymbol{n} = -50 \ m^3/day$ at the inlet, and $S_w = 0.2$, $P_w = 0$ at the outlet. The simulation is run for 200 days with time step $\Delta t = 20$ days.

For capillary pressure models, we employ a simple linear model and the Brooks-Corey [20] model:

$$\text{Linear model: } P_c(S_w) = P_0(1 - \bar{S}_w), \quad \text{Brooks-Corey: } P_c(S_w) = P_d \bar{S}_w^{-1/\lambda}, \quad (3.27)$$

in which $\bar{S}_w$ is the effective saturation, P_d is the entry pressure, and λ is related to the pore-size distribution. For the Brooks-Corey model, the typical range of λ is $[0.2, 3.0]$ [9,36]. In general, λ is greater than 2 for narrow distributions of pore sizes, and λ is less than 2 for wide distributions. For example, sandpacks with broader distributions of particle sizes have λ ranging from 1.8 to 3.7 [20]. The Brooks-Corey capillary pressure curves for various values of λ are plotted in figure 3.6. Other parameters are listed in table 3.2 and example 1 of table 3.3.

For all of the simulations presented here, the convergence tolerance for Newton's method is $||F(x)|| \leq 10^{-6}$, and the linear tolerance for GMRES is $||J\delta u_k - F(u_k)|| \leq 10^{-12}||F(u_k)||$, which is the default in Amanzi. BoomerAMG is used as a preconditioner. The number of V-cycle steps is set to 1. The coarsening strategy is

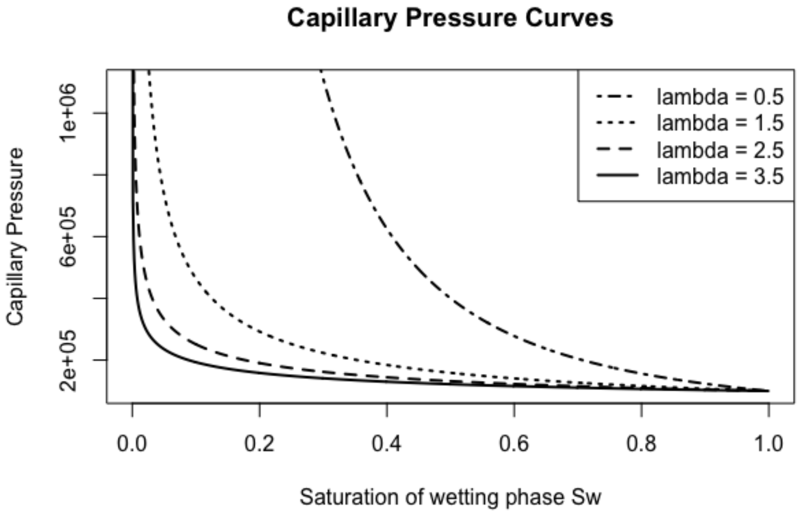


Figure 3.6: Capillary pressure curves for Brooks-Corey model with entry pressure $P_d = 10^5 \ \ Pa$.

the parallel Cleary-Luby-Jones-Plassman (CLJP) coarsening [32]. The interpolation method is the classical interpolation defined in [80], and the smoother is the forward hybrid Gauss-Seidel / successive over-relaxation (SOR) scheme.

In order to explore the effects of different models for capillary pressure on solver performance, we use the four sets of parameters listed in table 3.3. In Example 1, the parameters are chosen such that the L_∞ norm of the derivative of capillary pressure P_c' is large, leading to a diffusion-dominated case (see equation (3.12)). In Example 2, the parameters are tuned to reduce the L_∞ norm of P_c', leading to an advection-dominated case. Example 3 is a more extreme case of example 2, in which P_c' is further decreased, leading to a strongly advection-dominated case. Example 4 represents another diffusion-dominated case, and it is only used in the scaling test in section 4.5. We also note the difference between the linear model

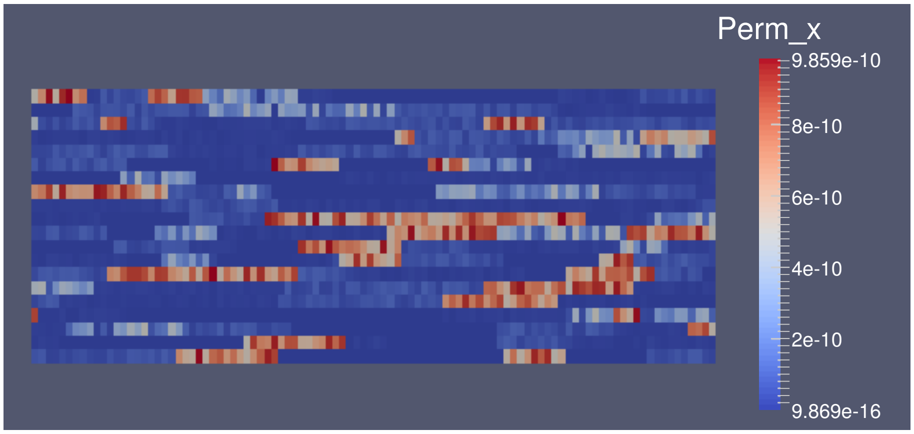


Figure 3.7: Permeability field obtained from SPE10 model 1 data.

The x-direction is scaled down by 1/20 for visualization.

Initial wetting phase pressure	$10^5\ Pa$
Initial nonwetting phase saturation	0.8
Residual wetting phase saturation	0.0
Nonwetting phase density	$700\ kg/m^3$
Wetting phase density	$1000\ kg/m^3$
Nonwetting phase viscosity	$10.0\ cP$
Wetting phase viscosity	$1\ cP$
Porosity	0.2

Table 3.2: Input data for the quarter-five spot problem.

Parameters	Ex 1	Ex 2	Ex 3	Ex 4
Linear entry pressure P_0	10^5	10^4	10^3	10^6
Brooks-Corey entry pressure P_d	10^6	10^5	2×10^4	10^6
Brooks-Corey λ	2.5	0.8	2.5	0.8

Table 3.3: Parameters for capillary pressure models

and the Brooks-Corey model for capillary pressure. The derivative P_c' for the linear model is a constant value, which means that the character of the problem, i.e. diffusion-dominated or advection-dominated, is the same everywhere for the whole domain. In the Brooks-Corey model, P_c' depends on the saturation of the wetting phase, and the problem can be diffusion-dominated in one part of the domain, and advection-dominated in another part. This can cause further difficulties for AMG-based solvers, whose optimal performance is sensitive to the characteristics of the problem.

The performance of the three strategies is summarized in tables 3.4 to 3.6. NI denotes the number of nonlinear iterations, LI the number of linear iterations, LI/NI the average number of linear iterations per nonlinear iterations, and Time the total time in seconds of the whole simulation. For the diffusion-dominated problem for which the results are shown in table 3.4, AMG is the most efficient method, about 25% more efficient than the block preconditioner in terms of both iteration counts (linear iterations per Newton step) and total run time. Note that in this example, the diffusion term in the (2,2)-block (A_{ss}) is large and the block is close

Methods/Models	Linear				Brooks Corey			
	NI	LI	LI/NI	Time	NI	LI	LI/NI	Time
AMG	32	368	11.5	27.2	36	470	13.1	37.24
CPR-AMG(1)	32	3695	115.5	324.15	36	5831	162	567.7
CPR-AMG(2)	32	899	28.1	103.94	36	1102	30.6	134.6
BF	32	524	16.4	33.17	36	599	16.6	46.2

Table 3.4: Performance of three preconditioning strategies for set of parameters in the diffusion-dominated Example 1.

to a scalar elliptic problem. Hence, it is not surprising that AMG performs well in this case. For the linear model, the block factorization approach still takes about 8 times fewer linear iterations, and it is about 10 times faster in total run time than CPR-AMG(1). The reason for this discrepancy is that CPR-AMG(1) is a two-stage preconditioner, and it requires an extra global solve using ILU. The block factorization preconditioner does not rely on ILU, which helps improve the run time significantly. CPR-AMG(2) also performs well in this case. Although it requires one more AMG solve per Newton iteration than CPR-AMG(1), it still outperforms CPR-AMG(1) in terms of both the number of linear iterations per Newton step and the total run time. The same conclusion can be made for the Brooks-Corey model.

The results reported in table 3.5 reveal the lack of robustness of AMG when applied to the coupled system. In contrast to the diffusion-dominated case, for

Methods/Models	Linear				Brooks Corey			
	NI	LI	LI/NI	Time	NI	LI	LI/NI	Time
AMG	37	2575	69.6	138.8	-	-	-	-
CPR-AMG(1)	37	1919	51.9	175.5	55	4851	88.2	605.7
CPR-AMG(2)	37	1222	33.0	157.1	55	3701	67.3	506.8
BF	37	684	18.5	51.7	55	1633	29.7	131.1

Table 3.5: Performance of three preconditioning strategies for set of parameters in the advection-dominated Example 2.

Methods/Models	Linear				Brooks Corey			
	NI	LI	LI/NI	Time	NI	LI	LI/NI	Time
AMG	-	-	-	-	-	-	-	-
CPR-AMG(1)	43	1079	25.1	122.8	48	2173	45.3	247.6
CPR-AMG(2)	43	1442	35.5	169.8	48	4805	100.1	560.5
BF	43	1002	23.3	69.8	48	1829	38.1	121.8

Table 3.6: Performance of three preconditioning strategies for set of parameters in the strongly advection-dominated Example 3.

the linear model of capillary pressure, AMG requires the highest number of linear iterations per Newton step for the advection-dominated case, and it even diverges for the Brooks-Corey model. The block factorization preconditioner still shows good performance, taking about half the number of iterations and running four times faster than the next best method, which is CPR-AMG(2). CPR-AMG(1) is still the least effective method in this case for both capillary pressure models.

For the strongly advection-dominated problem with parameters in example 3 (see table 3.6), AMG diverges for both the linear and Brooks-Corey capillary pressure models. The performance of CPR-AMG(2) is also affected in this case, trailing that of CPR-AMG(1). CPR-AMG(2) is still more robust than direct application of AMG, however, since unlike AMG, this method still converges. The block factorization preconditioner is again the most effective method, requiring fewer number of iterations and about half the run time of CPR-AMG(1). This suggests that when the diffusion term in the A_{ss} block gets small, the coupling block A_{ps} which has the structure of a discrete first-order hyperbolic problem for the saturation, becomes important and needs to be taken into account. The block factorization method does exactly this. Moreover, it takes advantage of the effectiveness of AMG for scalar problems. Recall that in the block factorization approach, AMG is applied to the approximate Schur block $\tilde{S}$, and to the block A_{ss}. $\tilde{S}$ has the form of a perturbed elliptic problem, and therefore we believe it is similar in character to the original pressure block A_{pp}. Thus, AMG is a natural choice for approximating the action of the inverse of $\tilde{S}$. Similarly, the block A_{ss} is a discrete version of a advection-diffusion problem, for which AMG should work well.

Methods/Mesh sizes	20^2	40^2	80^2	160^2
AMG	7	7	7	7
CPR-AMG(1)	15.1	25.9	49.4	95.2
CPR-AMG(2)	22.0	30.9	38.1	40.7
BF	19.9	21.0	21.1	21.1

Table 3.7: Iteration counts for diffusion-dominated case with gravity, time step $dt = 10$ days.

3.4.2 Two-dimensional problem with gravity

In this example, we compare the performance of the different strategies for a problem in which gravity plays a dominant role. The domain is a square box of size 20 × 20 meters. The absolute permeability is a homogeneous field of 100 millidarcy. Water is injected into the domain through the boundary at the top left corner, and the outlet is at the top right corner. The rate of injection is 5 m^3/day. For spatial discretization, we use uniform grids of size 20 × 20, 40 × 40, 80 × 80, and 160 × 160 respectively. The initial conditions are the same as the heterogeneous two-dimensional example above. The time steps are 10, 4, and 1 days, and the final times are 20, 8, and 2 days respectively.

The diffusion-dominated case, shown in table 3.7, exhibits the same pattern as in the previous example: the AMG preconditioner is the most efficient method, followed by the block factorization method, CPR-AMG(2), and CPR-AMG(1). AMG

Methods/Mesh sizes	20^2	40^2	80^2	160^2
AMG	-	-	-	17.3
CPR-AMG(1)	17.9	17.8	16.1	25.2
CPR-AMG(2)	30.3	29.8	22.7	30.2
BF	13.0	17.0	21.1	23.9

Table 3.8: Iteration counts for advection-dominated case with gravity, time step $dt = 4$ days.

Methods/Mesh sizes	20^2	40^2	80^2	160^2
AMG	-	-	-	-
CPR-AMG(1)	18.6	19.6	19.7	18.8
CPR-AMG(2)	31.5	34.6	36.6	36.3
BF	13.1	9.3	12.0	16.4

Table 3.9: Iteration counts for highly advection-dominated case with gravity, time step $dt = 1$ day.

and the block factorization method exhibit optimal performance with respect to problem size. The number of iterations for CPR-AMG(2) also seems to reach a plateau as the mesh size is refined. In contrast, the performance of CPR-AMG(1) does not scale well with respect to mesh size for this case, taking about twice the number of iterations for each level of mesh refinement.

The results for the advection-dominated case are shown in table 3.8. The AMG method is not robust and only converges for the largest mesh size (for which it takes the fewest iterations). The block factorization preconditioner is highly robust and also appears to require iteration counts tending to a constant as the mesh is refined. The performance of CPR-AMG(2) is consistent except for the 80×80 mesh. Although it requires more iterations than CPR-AMG(1), this method shows promising scaling property, similar to the previous example, since the number of iterations does not grow as the mesh is refined. CPR-AMG(1) performs quite well for this case, but it still exhibits poor scalability as the number of iterations grow quickly between 80×80 and 160×160.

In the strongly advection-dominated case, AMG diverges for all mesh sizes. The new block factorization is the most efficient method in this case, requiring the smallest number of iterations across all mesh sizes. Here, CPR-AMG(1) is more efficient than CPR-AMG(2), requiring about half the number of iterations. Both CPR-AMG(1) and CPR-AMG(2) show good scaling property in this case. The scaling result for the block factorization method is not as clear as in the diffusion and advection dominated cases, but we suspect that the mesh is not fine enough for a consistent pattern to emerge.

Besides varying the mesh size, we also experimented with changing the time step size for a fixed mesh of 80×80 for the same problem. The final time for the simulation is 8 days. The results are reported in table 3.10. Since AMG does not converge in this experiment, we exclude it from the results. It is clear that as the time step gets larger, Newton's method takes more iterations to converge. For $dt =$ 8 days, there is only one time step and it is the most difficult case. CPR-AMG(1) number of iterations is not significantly affected by the time step except for the largest time step size of 8 days. Meanwhile, CPR-AMG(2) number of iterations decreases as the time step gets larger, but goes up again at $dt = 8$ days. The block factorization method shows consistent increase in the number of iterations for larger time steps. Overall, there is not much of a difference in terms of iteration counts for these three methods, but it is worth noting that the block factorization method is much faster than the others in terms of run time, as it does not require a global ILU solve.

3.4.3 Behavior of Eigenvalues

It is often possible to obtain insight into the properties of preconditioning operators from the eigenvalues of the preconditioned matrix JM^{-1}. In particular, recall a standard analysis of the convergence behavior of GMRES for solving the preconditioned system (equation (3.16)) [83]. Assume the preconditioned matrix is diagonalizable, $JM^{-1} = V\Lambda V^{-1}$, where Λ is a diagonal matrix containing the eigenvalues of the preconditioned matrix and the columns of V are the corresponding

Methods/Time steps	$dt = 1$ day		$dt = 2$ days		$dt = 4$ days		$dt = 8$ days	
	NI/TS	LI/NI	NI/TS	LI/NI	NI/TS	LI/NI	NI/TS	LI/NI
CPR-AMG(1)	12.4	16.9	17	15.9	23	16.1	28	19.4
CPR-AMG(2)	12.4	29.0	17	24.0	23	22.7	28	27.2
BF	12.4	16.7	17	19.0	23	21.1	28	23.0

Table 3.10: Results for the advection-dominated case with gravity $P_0 = 10^4$.

NI/TS is the number of Newton iteration per time step.

eigenvectors. If $c_k = M^{-1}\hat{c}_k$ are the iterates obtained at the kth step of GMRES iteration, with residual $r_k = q - Jc_k$, then

$$\frac{\|r_k\|_2}{\|r_0\|_2} \leq \|V\|_2 \, \|V^{-1}\|_2 \min_{p_k(0)=1} \max_{\lambda \in \sigma(JM^{-1})} |p_k(\lambda)|, \tag{3.28}$$

where the minimum in equation (3.28) is over all polynomials of degree at most k that have the value 1 at the origin, $\sigma(JM^{-1})$ is the set of eigenvalues of JM^{-1}, and the norm is the vector Euclidian norm. Thus, a good preconditioner tends to produce a preconditioned operator with a compressed spectrum whose entries are not near the origin. In this section, we explore the behavior of the eigenvalues of the preconditioned matrix with an eye toward understanding the effects of features of the discrete problem such as discretization mesh size and qualitative features of the model such as the relative weights of diffusion and advection and the degree of coupling between the components.

Figure 3.8 gives a representative depiction of the eigenvalues of preconditioned operators for three of the preconditioners considered. These results are for benchmark problems for which performance is considered in section 3.4.1, the two-dimensional linear oil-water model discretized on a 100 × 20 grid. The plots on the left side of the figure show eigenvalues for the diffusion-dominated case (for which solution performance is shown in table 3.4), and those on the right show eigenvalues for the advection-dominated case (performance in table 3.5). [2]

These displays indicate that the spectra for the preconditioned systems for the CPR-AMG(2) and BF preconditioners are bounded away from the origin, whereas for the CPR-AMG(1) preconditioner there are many small eigenvalues. Performance of CPR-AMG(1) improves in the advection-dominated case, and the smallest associated eigenvalues are somewhat further from the origin. In contrast, the latter two preconditioners are largely unchanged in the advection-dominated case, where they are still effective, and the associated eigenvalues are also contained in similarly structured regions far from the origin. We believe the superior performance of the BF preconditioner comes from its greater emphasis on the coupling between pressure and saturation, derived from use of the approximate Schur complement (equation (3.22)).

[2] These computations were done using the `eig` function in Matlab, and they use Matlab backslash to perform the actions of the inverses of A_{11}, A_{22} and the modified Schur complement. This contrasts with the solution algorithms tested, which approximate these operations using one AMG V-cycle.

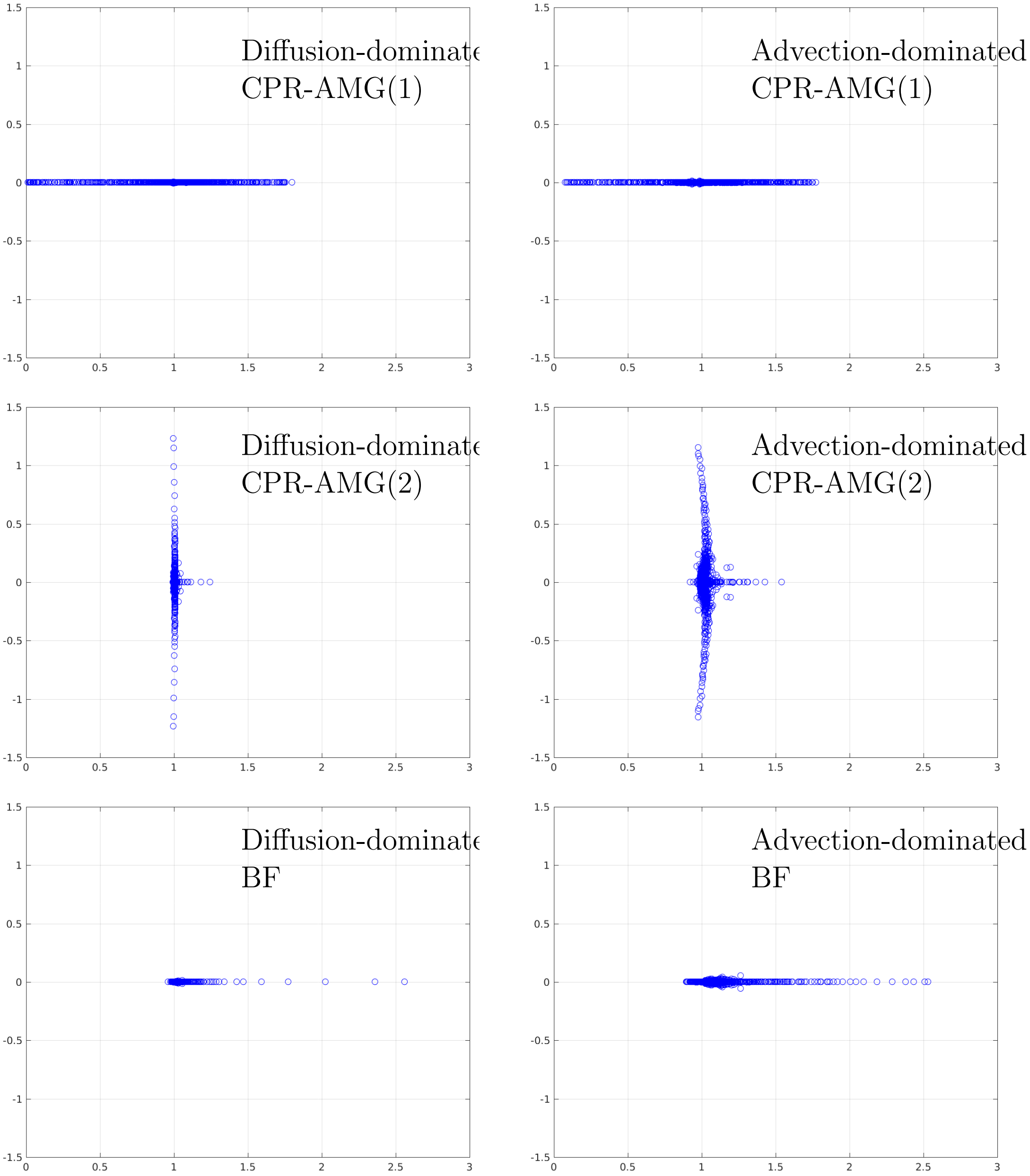


Figure 3.8: Eigenvalues of preconditioned systems for different strategies, applied to the diffusion-dominated Example 1 (left) and advection-dominated Example 2 (right).

Methods/Models	Linear				Brooks Corey			
	NI	LI	LI/NI	Time	NI	LI	LI/NI	Time
AMG	16	282	17.6	103.1	20	452	22.6	144.7
CPR-AMG(1)	16	2698	168.6	803.2	20	6069	303.45	1940.8
CPR-AMG(2)	16	712	44.5	299.5	20	1900	95.0	741.1
BF	16	355	22.2	133.6	20	752	37.6	231.1

Table 3.11: Performance in the 3D case for the set of parameters in example 1 of table 3.3.

3.4.4 Three-dimensional Problem

We use a homogeneous permeability field of 100 millidarcy, and the grid is stretched to induce anisotropy. The model dimensions are 25 × 100 × 6 meters and the cell size is 0.5 × 1 × 0.05 meter. Thus, the mesh is 50 × 100 × 120, and the problem has 1.2 million unknowns in total. Water is injected into the domain at one bottom corner and the outlet is at the opposite corner. The injection rate is 0.75 m^3/day. The parameters for the capillary pressure model is from example 1 of table 3.3. The simulation is run for 100 days with time step $\Delta t = 20$ days. Table 3.11 shows the performance results of the diffusion-dominated case for this 3D example, which are consistent with those of the previous two-dimensional example. AMG preconditioner shows the best results for both the iteration counts per Newton step and the time it takes to complete the simulation for both capillary pressure models.

CPR-AMG(2) does not perform quite as well as AMG, but it is much more efficient than CPR-AMG(1) for both performance measures and capillary pressure models. As in the two-dimensional case, the new block factorization method performs well, requiring fewer than half the iterations than CPR-AMG(2) for both the linear and Brooks-Corey models, and running in about one third the CPU time.

We also tested the three-dimensional SPE10 problem with the linear model of capillary pressure for the different preconditioning strategies. Here, AMG diverges even for the diffusion-dominated case ($P_0 = 10^6$ Pa), even though it was the most efficient method for the two-dimensional example. The block factorization method is about four times faster than CPR-AMG(2) and five times faster than CPR-AMG(1) in the diffusion dominated case. CPR-AMG(2) still outperforms CPR-AMG(1) both in terms of iteration counts and run time, but the margin is smaller than for the two-dimensional problem (table 3.12). In the advection-dominated case ($P_0 = 10^5$ Pa), unlike in the two-dimensional example, CPR-AMG(1) is more efficient than CPR-AMG(2), requiring about 23% fewer number iterations and 45% run time. The block factorization approach is still the most efficient method, taking fewer than half the number of iterations and less than half the run time of CPR-AMG(1) (table 3.13). We also note that the number of iterations for the block factorization method is very consistent with respect to the characteristics of the problem, i.e. it does not change significantly whether the problem is diffusion-dominated, advection-dominated, or strongly advection-dominated.

Methods/Models	Linear			
	NI	LI	LI/NI	Time (s)
AMG	-	-	-	-
CPR-AMG(1)	17	2410	141.8	614.14
CPR-AMG(2)	17	1661	97.7	448.21
BF	17	490	28.8	121.71

Table 3.12: Performance for the three-dimensional SPE10 model, diffusion-dominated case.

Methods/Models	Linear			
	NI	LI	LI/NI	Time (s)
AMG	-	-	-	-
CPR-AMG(1)	18	1122	62.3	354.38
CPR-AMG(2)	18	1554	86.3	657.12
BF	18	474	26.3	157.24

Table 3.13: Performance for the three-dimensional SPE10 model, advection-dominated case.

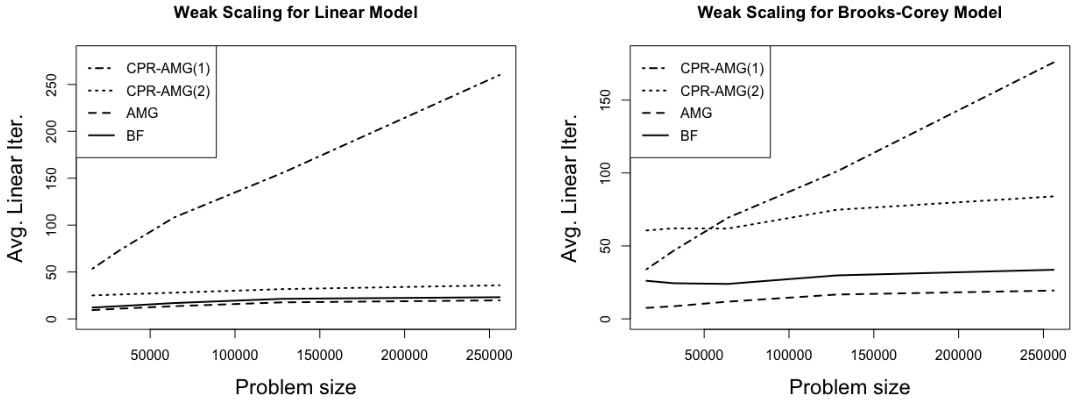


Figure 3.9: Weak scaling for different strategies.

3.4.5 Scaling Results

To perform a scalability study, we run a test problem on a box of dimensions $20 \times 20 \times 20$ meters. The initial mesh is $20 \times 20 \times 20$ and is repeatedly refined in the z-direction. The time step is fixed at $\Delta t = 20$ days. The domain has constant material properties. The parameters for the capillary pressure models are listed in example 4 of table 3.3. Note that this set of parameters corresponds to a diffusion-dominated problem. The results shown in figure 3.9. indicate that the performance of the block factorization, CPR-AMG(2), and AMG methods is independent of the mesh size. The number of linear iterations per Newton step does not grow as the mesh is refined which is optimal multigrid performance. The block factorization method's performance is nearly identical to that of AMG for the linear model, and still quite close for the Brooks-Corey model, compared to CPR-AMG(2). CPR-AMG(1), however, does not scale as well as the other two methods. The linear iteration counts for CPR-AMG(1) grows linearly as the mesh is refined.

3.5 Conclusions

In this chapter, we showed the performance of our implementation of a fully implicit parallel isothermal two-phase flow simulator along with four different preconditioning strategies to solve the linear systems resulting from linearization of the coupled equations, and we have tested the performance of these methods as preconditioners for GMRES. We have also developed a new block factorization preconditioner whose performance is robust and efficient across all benchmark problems studied. In contrast, although AMG preconditioning applied to the coupled systems is the most efficient choice in some cases (both two-dimensional and three-dimensional diffusion-dominated examples), it exhibits slow convergence and sometimes diverges for advection-dominated cases. The new block factorization preconditioner achieves consistently low iteration counts across all the tests and varying examples of capillary pressure, and it scales optimally with problem size. The combinative CPR-AMG(1), though robust across all the tests, is the least efficient method, with the exception of the near hyperbolic case where it is faster than CPR-AMG(2). The additive CPR-AMG(2) method performs well in most cases except the strongly advection-dominated case. It also scales optimally with problem size for both advection-dominated and diffusion-dominated case.

Chapter 4: Multigrid Reduction Method for Multiphase Flow in Porous Media with Phase Transitions

4.1 Introduction

The previous chapter is concerned with the development of scalable preconditioners for incompressible two-phase flow. This chapter focuses on the two-phase, two-component model with phase transitions. One important effect of phase transitions is that the governing system of PDEs can become degenerate, making the resulting linear systems indefinite. Again, Krylov subspace methods, such as the generalized residual method (GMRES) [84], can be applied to solve these systems. However, these methods by themselves generally converge slowly and they must be appropriately preconditioned to accelerate convergence. The ILU factorization is a popular approach as a preconditoner, due to its simplicity and robustness. However, as simulations cover larger and larger domains and are deployed over high performance parallel architectures, there is an apparent need for robust solvers that scale, and the use of standard (single-level) ILU methods becomes less favorable.

Previously, most of the research has focused on finding new formulations which can deal with phase transitions. Some approaches include primary variable switching (PVS) [51, 98], negative saturation [1], and finding a set of persistent primary variables [18, 63, 65]. Recently, a new approach has been developed for handling the phase transitions by formulating the system of equations as a nonlinear complementarity problem (NCP) [16, 59, 62]. Unlike the PVS approach, the advantage of the NCP approach is that the set of primary variables is consistent throughout the simulation, and no primary variable switching is needed. Not only is this approach more robust and efficient, it also presents an opportunity to use scalable linear solvers such as algebraic multigrid (AMG).

In this chapter, we develop a new family of AMG preconditioners based on multigrid reduction (MGR) for GMRES to solve the linear systems resulting from the discretization of the continuous problem. The algorithms and results presented here also appear in [22]. The MGR technique has been around for many years [78, 79]. It can be considered as a generalization of the multi-stage preconditioner in a standarfd multigrid framework. A closed form of the error propagator can be derived for the MGR approach, and this enables us to study the effect of different multigrid components on the convergence of the linear solver. In addition, the MGR framework has been shown to be an efficient preconditioner for different types of PDEs, such as models of reservoir simulation, and it has also been applied with varying degree of success to the time dimension [47].

We consider a two-phase, two-component system with phase transitions as our model problem. We describe this model in detail in section 4.2. Classical approaches to simulate two-phase, two-component are well-posed if two primary variables are chosen in advance. For example, one can choose one phase pressure and one phase saturation, or one phase pressure and one component concentration. This set of variables remains fixed in the case of phase appearance/disappearance if one knows in advance what phase will appear and disappear during the simulation. Then a constrained pressure residual (CPR) preconditioning approach [38, 74, 93] can be employed to obtain a semi-elliptic pressure equation. The pressure equation is solved with a multi-level method, such as AMG or multiscale [37], and this is followed by a relaxation step with ILU for the global linear system. Although this approach has been shown to be very effective for some real world examples [37], it is not always robust in cases with strong capillarity effects [21]. Due to the fact that the CPR approach uses ILU in the smoothing step, it also may not scale as well as a block factorization approach [21]. The goal of this chapter is to develop a new multigrid algorithm that is both robust, efficient, and also general enough to accommodate various formulations of compositional multiphase flow. In particular, we show that, under appropriate assumptions, our multigrid reduction method is equivalent to the CPR-AMG and block factorization approaches.

The rest of this chapter is organized as follows: In section 4.3, we describe the fully implicit discretization. We briefly review the MGR framework in section 4.4 and explain our new MGR algorithm in section 4.5. In section 4.6, several numerical tests are used to assess the robustness and scalability of the new algorithm. Some concluding remarks as well as future work are presented in section 4.7.

4.2 Problem Statement

4.2.1 Governing Equations

We consider a simplified two-phase two-component model with phase transitions, similar to that presented in [19]. This model provides a simple example that demonstrates the capability of the nonlinear complementarity constraint approach to handle phase appearance and disappearance. The flow consists of gas and liquid phases, and the components are hydrogen and water. We make the following simplifications: (1) water does not vaporize so the gas phase contains only hydrogen, and (2) the amount of hydrogen dissolved into the liquid phase is small. For a complete set of assumptions, we refer to [18]. For the two components, the mass conservation equations read

$$\phi \frac{\partial(\rho_l^w S_l)}{\partial t} + \nabla \cdot (\rho_l^w \boldsymbol{q}_l - j_l^h) = 0, \tag{4.1}$$

$$\phi \frac{\partial(\rho_l^h S_l + \rho_g^h S_g)}{\partial t} + \nabla \cdot (\rho_l^h \boldsymbol{q}_l + \rho_g^h \boldsymbol{q}_g + j_l^h) = 0, \tag{4.2}$$

where the subscripts l, g denote the liquid and gas phases, and the superscripts w, h denote the water and hydrogen components, respectively. The porosity of the medium is denoted ϕ, $Sat\alpha, \boldsymbol{q}_\alpha$ are the saturation and velocity of phase α, respectively; ρ_l^h is the dissolved hydrogen mass concentration in the liquid phase; and j_l^h is the diffusion flux of hydrogen in the liquid phase. The Darcy velocity $\boldsymbol{q}_\alpha$ follows the Darcy-Muskat law

$$\boldsymbol{q}_\alpha = -\boldsymbol{K} \lambda_\alpha \nabla (P_\alpha - \rho_\alpha \boldsymbol{g}), \quad \alpha = l, g, \tag{4.3}$$

where K is the absolute permeability, λ_α, P_α, and ρ_α are the mobility, pressure, and density of phase α, and $\boldsymbol{g}$ is the gravitational acceleration. The mobility λ_α of phase α is defined as the ratio between the phase relative permeability $k_{r\alpha}$ and the phase viscosity μ_α: $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$. Using Fick's law, the diffusion flux of hydrogen in liquid j_l^h in equations (4.1) and (4.2) can be expressed as

$$j_l^h = -\phi S_l D_l^h \nabla \rho_l^h, \tag{4.4}$$

where D_l^h is hydrogen molecular diffusion coefficient in liquid. Since we assume incompressibility of the liquid phase, the mass density of the water component in the liquid phase is constant, i.e. $\rho_l^w = \rho_w^{std}$. To capture capillarity effects, the jump in the pressure at the interface of the two phases is modeled by the relation

$$P_g = P_l + P_c(S_l). \tag{4.5}$$

where P_c is the capillary pressure. Additionally, we have the constraints

$$S_l + S_g = 1. \tag{4.6}$$

To close the model, we also need a set of equations for the thermodynamic equilibrium. Neglecting water vapor and assuming low solubility of hydrogen in the liquid phase, Henry's law can be used to connect the gas pressure P_g and the dissolved hydrogen mass concentration in liquid ρ_l^h:

$$\rho_l^h = C_h P_g, \tag{4.7}$$

where $C_h = HM^h = \rho_w^{std} M^h / M^w$, H is the Henry's law constant, and M^k, $k \in \{w, h\}$ are the molar mass of the k-th component. Again, since we ignore water vapor in the gas phase, the ideal gas law reads

$$\rho_g^h = \rho_g = C_v P_g, \tag{4.8}$$

where C_v is a constant and $C_v = M^h/(RT)$; T is the temperature and R the ideal gas constant.

4.2.2 Nonlinear Complementarity Problem

To handle phase transitions, we introduce the following nonlinear complementarity problem

$$C_h P_g - \rho_l^h \geq 0, \quad 1 - S_l \geq 0, \quad (1 - S_l)(C_h P_g - \rho_l^h) = 0. \tag{4.9}$$

Equivalently, we can rewrite the above equation using the min function as in [16,59]

$$\min(1 - S_l, C_h P_g - \rho_l^h) = 0. \tag{4.10}$$

Although one can use other types of complementarity functions, the min function is convenient because of its piecewise linearity with respect to the variable S_l and ρ_l^h, which simplifies the computation of the Jacobian in each nonlinear iteration. When the gas phase is not present, we have $C_h P_g - \rho_l^h > 0$ since $\rho_l^h = 0$, and equation (4.10) reduces to $1 - S_l = 0$. When the gas phase appears, $1 - S_l > 0$ and the constraint equation is governed by Henry's law in equation (4.7).

4.2.3 Relative Permeability Curves

In this work, we use two different models for relative permeability terms.

- Power law (Brooks-Corey type)

$$k_{rl} = S_{le}^2, \quad k_{rg} = (1 - S_{le})^2, \tag{4.11}$$

$$S_{le} = \frac{S_l - S_{lr}}{1 - S_{lr} - S_{gr}}. \tag{4.12}$$

- Van Genuchten law [91]

$$k_{rl} = \sqrt{S_{le}}\Big(1 - \big(1 - S_{le}^{1/m}\big)^m\Big)^2, \tag{4.13}$$

$$k_{rg} = \sqrt{1 - S_{le}}\Big(1 - S_{le}^{1/m}\Big)^{2m}, \tag{4.14}$$

$$m = 1 - \frac{1}{n}, \tag{4.15}$$

where S_{le} is the effective liquid saturation, and $S_{lr}, S_{gr} \in [0, 1]$ are the residual saturations of the liquid and gas phase, respectively. As we have mentioned in section 2.2.2, the Van Genuchten model is better suited for fine-textured soils and dense materials, whereas the Brooks-Corey model is more accurate for coarse-textured soils. Since the goal of this chapter is to develop a preconditioner that works well across a wide range of porous media with different soil characteristics, it is important that we experiment with both of these models.

4.2.4 Capillary Pressure

We employ two models for capillary pressure

- Linear model

$$P_c = P_r(1 - S_{le}). \tag{4.16}$$

- Van Genuchten model [91]

$$P_c = P_r\Big(S_{le}^{-1/m} - 1\Big)^{1/n}. \tag{4.17}$$

where P_r is the entry pressure. Notice that the function $P_c(S_l)$ in the Van Genuchten model is only defined for $S_l \in [S_{lr}, 1-S_{gr}]$ and P_c' is unbounded near S_{lr} and $1-S_{gr}$. Thus, it is necessary to modify the model to limit the growth of P_c' and extend it for $S_l \in \mathbb{R}$, since the value of S_l can become larger than $1 - S_{gr}$ or less than S_{lr} during the nonlinear iteration. We used a regularization as presented in [62] with parameter $\epsilon = 10^{-5}$.

- For $S_l \in [S_{lr}, 1 - S_{gr}]$

$$\tilde{S} := S_{lr} + (1-\epsilon)(S_l - S_{lr}) + \frac{\epsilon}{2}(1 - S_{gr} - S_{lr}) \tag{4.18}$$

$$\tilde{P}_c(S_l) = P_c(\tilde{S}) - P_c\Big(S_{lr} + \frac{\epsilon}{2}(1 - S_{gr} - S_{lr})\Big) \tag{4.19}$$

- For $S_l < S_{lr}$

$$\tilde{P}_c(S_l) = \tilde{P}_c(S_{lr}) + \tilde{P}_c'(S_{lr})(S_l - S_{gr}) \tag{4.20}$$

- For $S_l > 1 - S_{gr}$

$$\tilde{P}_c(S_l) = \tilde{P}_c(1 - S_{gr}) + \tilde{P}_c'(1 - S_{gr})(S_l - 1 + S_{gr}) \tag{4.21}$$

In this regularization, for the saturation that is outside of the domain, capillary pressure is computed by a linear extrapolation from the regularization points $S_{lr} + \mathcal{O}(\epsilon)$ and $1 - S_{gr} - \mathcal{O}(\epsilon)$ with the slopes $P_c'(S_{lr} + \mathcal{O}(\epsilon))$ and $P_c'(1 - S_{gr} - \mathcal{O}(\epsilon))$, respectively.

4.2.5 Primary Variables

There are many ways to choose a set of primary variables, depending on the problem formulation and applications. In our model example, a convenient choice is the liquid saturation, liquid pressure, and the concentration of hydrogen in the liquid phase. We have our solution vector $u = \{P_l, S_l, \rho_l^h\}$. Unlike in other methods such as primary variable switching, in the NCP approach, the choice of primary variable is fixed throughout the simulation. This is an important feature for success of our multilevel algorithm discussed in section 4.5.

4.3 Solution Algorithm

Here, we consider solving the coupled system consisting of equations (4.1), (4.2) and (4.9) fully implicitly. We use a cell-centered finite volume method for spatial discretization, as it is a natural way to preserve the mass conservation property of the balance equations (4.1) and (4.2). In addition, it can deal with the case of discontinuous permeability coefficients, and it is relatively straightforward to implement. For the time domain, we again employ the backward Euler method to avoid the CFL stability restriction of the time step.

4.3.1 Semi-smooth Newton's Method

We want to solve the system

$$\boldsymbol{R}(u) = \begin{cases} \boldsymbol{H}(u) = 0 & \text{(from the PDEs)} \\ \\ \boldsymbol{\Theta}(u) = \min(\boldsymbol{F}, \boldsymbol{G}) = 0 & \text{(from the constraints)} \end{cases}$$

in which $\boldsymbol{F}$ and $\boldsymbol{G}$ are discrete functions of $1 - S_l$ and $C_g P_g - \rho_l^h$ respectively, and $\boldsymbol{R}(u)$ is the residual function. A straightforward approach for solving nonlinear systems of equations is the Newton's method, which requires solution of a linear system at each iteration k:

$$\left.\frac{\partial \boldsymbol{R}}{\partial u}\right|_{u=u_k} \delta u = -\boldsymbol{R}(u_k). \tag{4.22}$$

This method requires that the Jacobian $\partial \boldsymbol{R}/\partial u$ is defined everywhere. In the case of NCP formulation, the constraints $\boldsymbol{\Theta}$ are only differentiable almost everywhere, and we will need to consider a *semi-smooth* Newton's method instead. The procedure for the semi-smooth Newton's method is similar to that for Newton's method, except that we replace the derivative $\boldsymbol{\Theta}'$ with a member of the subdifferential $\partial\boldsymbol{\Theta}$ when the function $\boldsymbol{\Theta}$ is non-differentiable. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a locally Lipschitz-continuous function and D_F be the set where F is differentiable; the B-subdifferential of F at x is defined as the set

$$\partial_B F(x) := \left\{G \in \mathbb{R}^{n\times n} : \exists\, x_k \in D_F \text{ with } x_k \to x, \nabla F(x_k) \to G\right\}.$$

Algorithm 4 shows the semi-smooth Newton's method using the minimum function as the complementarity function as described in [16]. It is also referred to as the Newton-min algorithm. For our two-phase, two-component model, the active set A^k

Algorithm 4 Newton-min algorithm

```
while $k < max\_iter$ and $res > \epsilon$ do
    (1) Define the index sets $A^k$ and $I^k$:
    $A^k := \{j : \boldsymbol{F}_j(u^k) \geq \boldsymbol{G}_j(u^k)\}$, $I^k := \{j : \boldsymbol{F}_j(u^k) < \boldsymbol{G}_j(u^k)\}$
    (2) Select an element $J^k \in \partial\boldsymbol{\Theta}(u^k)$ such that its $j$th row is equal to
    $\boldsymbol{F}'_j(u^k)$ if $j \in I^k$, $\boldsymbol{G}'_j(u^k)$ if $j \in A^k$
    (3) Solve the system
    $\boldsymbol{H}'(u^k) \triangle u^k = -\boldsymbol{H}(u^k)$
    $J^k \triangle u^k = -\boldsymbol{\Theta}(u^k)$
    (4) Update $u^{k+1}$
    $u^{k+1} = u^k + \triangle u^k$
```

corresponds to the set of last rows where the gas phase is present. The general semi-smooth Newton's method converges locally superlinearly for semi-smooth functions, and quadratically for strongly semi-smooth functions. Definitions of semi-smooth and strongly semi-smooth are given in [72], and a complete treatment of the semi-smooth Newton's method with active set strategy is presented in [54].

The linear system resulting from taking the subdifferential $\partial\boldsymbol{R}/\partial u$ is often very difficult to solve using iterative methods, and preconditioning is critical for rapid convergence of Krylov subspace methods such as GMRES. In the next section, we discuss the linear system arising from the semi-smooth Newton's method and give a detailed description of the solution algorithms we will use to solve this system.

4.3.2 Linear System

Assuming that each physical variable is ordered lexicographically, then each nonlinear iteration entails the solution of a discrete version of a block linear system of the form

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}, \tag{4.23}$$

in which the matrices in the first two rows are the discretized version of the linearized operators from the PDEs. Let $\delta P_l, \delta S_l, \delta \rho_l^h$ be the updates for pressure, saturation, and hydrogen density at each nonlinear step. Using Taylor expansion and keeping only the linear terms, we have

$$A_{11} = -\nabla \cdot (\rho_l^w \boldsymbol{K} \lambda_l \nabla \delta P_l),$$

$$A_{12} = \phi \frac{\partial}{\partial t}(\rho_l^w \delta S_l) - \nabla \cdot (\rho_l^w \boldsymbol{K} \lambda_l' \nabla \tilde{P}_l \delta S_l) + \nabla \cdot (\phi D_l^h \nabla \tilde{\rho}_l^h \delta S_l),$$

$$A_{13} = \nabla \cdot (\phi S_l D_l^h \nabla \delta \rho_l^h),$$

$$A_{21} = \phi \frac{\partial}{\partial t}(S_g C_g \delta P_l) - \nabla \cdot (\rho_l^h \boldsymbol{K} \lambda_l \nabla \delta P_l) - \nabla \cdot (\rho_g^h \boldsymbol{K} \lambda_g \nabla \delta P_l) - \nabla \cdot (C_g \boldsymbol{K} \lambda_g \nabla P_g \delta P_l),$$

$$A_{22} = \phi \frac{\partial}{\partial t}((\rho_l^h - \rho_g^h) \delta S_l) - \nabla \cdot (\rho_l^h \boldsymbol{K} \lambda_l' \nabla P_l \delta S_l) - \nabla \cdot (\rho_g^h \boldsymbol{K} \lambda_g' \nabla P_g \delta S_l)$$

$$- \nabla \cdot (C_g P_c' \boldsymbol{K} \lambda_g \nabla P_g \delta S_l) - \nabla \cdot (\rho_g^h \boldsymbol{K} \lambda_g \nabla P_c' \delta S_l) - \nabla \cdot (\rho_g^h \boldsymbol{K} \lambda_g P_c' \nabla \delta S_l)$$

$$- \nabla \cdot (\phi D_l^h \nabla \rho_l^h \delta S_l),$$

$$A_{23} = \phi \frac{\partial}{\partial t}(\tilde{S}_l \delta \rho_l^h) - \nabla \cdot (\phi \tilde{S}_l D_l^h \nabla \delta \rho_l^h).$$

All the coefficients in the above equations are evaluated at the linearization point $\{\tilde{P}_l, \tilde{S}_l, \tilde{\rho}_l^h\}$. From these operators, we can make some important observations:

- The global matrix is non-symmetric and indefinite.
- The block A_{11} has the structure of a discrete purely elliptic problem for pressure.
- The coupling block A_{12} has the structure of a discrete first-order hyperbolic problem in the liquid phase saturation.
- The coupling block A_{21} has the structure of a discrete parabolic problem in the wetting phase pressure.
- The block A_{22} has the structure of a discrete parabolic (convection-diffusion) problem for saturation when capillary pressure is a non-constant function of the saturation. When capillary pressure is zero or a constant, $P_c' = 0$ and there is no diffusion term, the block has the form of a hyperbolic problem.
- The entries of the blocks with respect to the dissolved hydrogen mass density A_{13}, A_{23} are small with respect to the diagonal block A_{11}, and only play a significant role in the regions where the gas phase does not exist.

These observations will help us motivate the development of our new method in the next section.

Besides the blocks associated with the PDEs, we also need to consider those in the last row of the matrix in equation (4.23), which are derived from the discrete version of the complementarity constraint (see equation (4.9)). When the gas phase does not exist, we have

$$A_{31} = 0, \quad A_{32} = -\delta S_l, \quad A_{33} = 0,$$

and when the gas phase is present, these blocks become

$$A_{31} = C_h \delta P_l,$$

$$A_{32} = C_h P_c' \delta S_l,$$

$$A_{33} = -\delta \rho_l^h.$$

In matrix form, the blocks A_{31}, A_{32}, and A_{33} are diagonal matrices, since the constraints are local. Again, because a phase can disappear, the block A_{33} is not guaranteed to be nonsingular. In fact, when this happens, the rows corresponding to the cells where a phase disappears have zero diagonal values. Thus, we can split the last row into two separate sets: the set with zeros on the diagonal of A_{33} and its complement. Rewriting the matrix A using this splitting we have:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ C_{31} & C_{32} & C_{33} & 0 \\ C_{41} & C_{42} & 0 & 0 \end{pmatrix}. \tag{4.24}$$

Let N be the number of elements in the mesh and M be the number of cells in which the gas phase is present. Then, the size of the matrix A is $3N \times 3N$. The blocks A_{ij}, $i = 1, 2$, $j = 1, 2$ are of dimension of $N \times N$. The pressure - hydrogen mass concentration A_{13}, A_{14} and saturation - hydrogen mass concentration A_{23}, A_{24} coupling blocks have dimension $N \times M$ and $N \times (N - M)$, respectively. The hydrogen mass concentration - pressure C_{31}, C_{41} and hydrogen mass concentration - saturation C_{32}, C_{42} constraint blocks have dimension $M \times N$ and $(N - M) \times N$, respectively. The block C_{33} is a diagonal matrix of size $M \times M$ which contains only non-zero diagonal values of A_{33}. The zero block on the diagonal has the size of $(N - M) \times (N - M)$. Since A has zeros on its diagonal, it is clear that we cannot use classical AMG algorithms to solve this system. In the past, since much of the focus was to find a formulation that can take into account all the complex physics involved in simulating compositional multiphase flow with miscibility and phase transitions, there has not been a lot of work in designing optimal preconditioners for this type of linear system. Recently, there have been some development of algebraic multigrid preconditioners such as two-stage preconditioning [87,94] and block factorization [21] for immiscible two-phase flow. Yet, these methods have not been applied successfully to the problems considered here. The most popular and robust method is still the incomplete factorization (ILU) of the global matrix A as a preconditioner for GMRES. In this work, we seek to develop a new algebraic multigrid preconditioner based on multigrid reduction for the linear system arising in the semi-smooth Newton's method.

4.4 Multigrid Reduction

The idea of multigrid reduction (MGR) has been around for a long time, tracing back to the work of Ries and Trottenberg [78, 79]. Recently, it has gained more attention through the work on multigrid reduction in time by Falgout et al. [47]. In this section, we summarize the approach for the case of two-level reduction. For a matrix A of size $N \times N$, we define a partition of the row indices of the matrix into C-points and F-points. The C-points play a role analogous to the points on a coarse grid, and the F-points belong to the set that is the complement of the C-points. It is important to note that this partitioning is different from the one usually used in standard multigrid methods, in which the F-points correspond to the points on the fine grid, i.e. the set of C-points is a subset of the set of F-points. In multigrid reduction, the C-points and F-points belong to non-overlapping sets. Using such a C-F splitting we have

$$A = \begin{pmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{pmatrix} = \begin{pmatrix} I_{ff} & 0 \\ A_{cf}A_{ff}^{-1} & I_{cc} \end{pmatrix} \begin{pmatrix} A_{ff} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I_{ff} & A_{ff}^{-1}A_{fc} \\ 0 & I_{cc} \end{pmatrix}, \tag{4.25}$$

where I_{cc} and I_{ff} are identity matrices and $S = A_{cc} - A_{cf}A_{ff}^{-1}A_{fc}$ is the Schur complement.

We can define the ideal interpolation and restriction operators by

$$P = \begin{pmatrix} -A_{ff}^{-1}A_{fc} \\ I_{cc} \end{pmatrix}, \quad R = \begin{pmatrix} -A_{cf}A_{ff}^{-1} & I_{cc} \end{pmatrix}. \tag{4.26}$$

Additionally, define the injection operator as $Q = \begin{pmatrix} I_{ff} & 0 \end{pmatrix}^T$. Then since $A_{ff} = Q^TAQ$ and $S = RAP$, it is simple to derive that

$$A^{-1} = P(RAP)^{-1}R + Q(Q^TAQ)^{-1}Q^T,$$

and

$$\begin{aligned} 0 = I - A^{-1}A &= I - P(RAP)^{-1}RA - Q(Q^TAQ)^{-1}Q^TA && (4.27) \\ &= (I - P(RAP)^{-1}RA)(I - Q(Q^TAQ)^{-1}Q^TA) && (4.28) \\ &= (I - Q(Q^TAQ)^{-1}Q^TA)(I - P(RAP)^{-1}RA), && (4.29) \end{aligned}$$

where the equivalence occurs since $RAQ = Q^T AP = 0$. This identity defines the two-level multigrid method with the ideal Petrov-Galerkin coarse-grid operator RAP and the F-relaxation $Q(Q^T AQ)^{-1}Q^T$. Equation (4.27) is the additive MGR identity and equations (4.28) and (4.29) are multiplicative identities with pre-smoothing and post-smoothing. However, constructing ideal interpolation and restriction operators is impractical, and we need to approximate these operators. In practice, MGR methods replace ideal restriction and prolongation with approximations R and P respectively, where

$$P = \begin{pmatrix} W_p \\ \\ I_{cc}, \end{pmatrix}, \quad R = \begin{pmatrix} W_r & I_{cc} \end{pmatrix}. \tag{4.30}$$

There are many ways to construct the restriction R and interpolation P operators. Here, we have only experimented with two options

$$W_r = 0, \quad W_p = -D_{ff}^{-1} A_{fc}, \tag{4.31}$$

and

$$W_r = -A_{cf} D_{ff}^{-1}, \quad W_p = -D_{ff}^{-1} A_{fc}, \tag{4.32}$$

where $D_{ff} = diag(A_{ff})$. The F-relaxation in equations (4.28) and (4.29) is also generally replaced with a more efficient method and often extended to all unknowns, not just F-points. We can solve with (block) Jacobi, (block) Gauss-Seidel, ILU, or AMG.

The coarse grid operator $A_c = RAP$ could also be considered as an approximation to the Schur complement. There are many proposed approximations to the Schur complement, including several based on multigrid ideas [8, 27, 75, 76, 82, 92].

Physics-based approximations for the Schur complement are of interest in general settings. There is a considerable literatures for saddle point problems [17,42,43,85]. Another interesting direction is based on the work on Block Factorized Sparse Approximate Inverse (Block FSAI) preconditioners in [50]. In the context of MGR, the Block FSAI could be used to construct good approximation to the restriction and interpolation operators.

In general, we define the MGR operator in either pre-smoothing or post-smoothing form by

$$I - M_{MGR}^{-1}A = (I - PM_c^{-1}RA)(I - M_f^{-1}A), \tag{4.33}$$

$$I - M_{MGR}^{-1}A = (I - M_f^{-1}A)(I - PM_c^{-1}RA), \tag{4.34}$$

where $M_c^{-1} \approx (RAP)^{-1}$ is the coarse grid correction and M_f^{-1} is the smoother. The two grid solve consists of an F-relaxation followed by a coarse-grid correction can be presented as follows

Algorithm 5 MGR preconditioner with presmoothing

```
Let $r = b$ and $e_0 = 0$.
(1) Global Relaxation:
$e_1 \leftarrow e_0 + M^{-1}r$, where $M = \text{blockdiag}(A)$
(2) F-Relaxation:
$e_2 \leftarrow e_1 + QM_{ff}^{-1}Q^T(r - Ae_1)$ where $M_{ff} = \text{blockdiag}(A_{ff})$
(3) Coarse Grid Correction:
$e_3 \leftarrow e_2 + PM_c^{-1}R_I(r - Ae_2)$ where $M_c^{-1}$ is approximated by a classical AMG method.
```

The appeal of the MGR approach is that it provides a flexible framework for choosing the coarse/fine grids, the interpolation and restriction operators, and the solver for the coarse/fine grids. For example, if one chooses to extend the F-points to all unknowns (rather than the complement of the C-points), $W_p = 0$, $W_r = 0$ for the interpolation and restriction operators, and ILU0 and AMG for the F-relaxation and coarse-grid solve, respectively, then the MGR method is equivalent to the CPR-AMG approach. The block factorization method in [21] is another variant of the MGR approach, which uses the C-points for the pressure and F-points for the saturation unknowns and $W_r = -A_{cf}D_{ff}^{-1}$, $W_p = 0$. Another advantage of the MGR approach is that it is an algebraic method and unlike geometric multigrid, it can be used as a "black-box" solver for general geometries and grid types.

So far we only assume that a C-F splitting of the rows is given. How to choose such a splitting is dependent on the problem and it is up to the user to make the decision. However, as a general principle, it is usually a good idea to assign the C-points to the variables associated with an elliptic equation, e.g. pressure, since we want to solve the coarse grid using an efficient method such as standard AMG. In the next section, we show how to choose an appropriate C-F splitting at each level of reduction for our particular problem. We also note that unlike BoomerAMG, which requires the matrix to be ordered by grid points (see section 3.3.4), our MGR method is designed for systems of PDEs, and it works with matrices in block form such as those in equations (4.23) and (4.24).

4.5 MGR for the two-phase, two-component model

Again, we need to solve the linear system $A\boldsymbol{u} = \boldsymbol{f}$, in which the matrix A is given in equation (4.24). The first step of multigrid reduction aims to eliminate the third row, corresponding to the constraints in the cells where all the phases exist. Thus, we have the following splitting:

$$A = \begin{array}{cc} \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ C_{31} & C_{32} & C_{33} & 0 \\ C_{41} & C_{42} & 0 & 0 \end{pmatrix} & \begin{array}{c} C \\ C \\ F \\ C \end{array} \end{array} \tag{4.35}$$

Note that the last column indicates the C-F splitting we use for this case. The Schur complement after the reduction step reads

$$S^1 = RAP = \begin{pmatrix} A_{11} & A_{12} & A_{14} \\ A_{21} & A_{22} & A_{24} \\ C_{41} & C_{42} & 0 \end{pmatrix} - \begin{pmatrix} A_{13} \\ A_{23} \\ 0 \end{pmatrix} C_{33}^{-1} \begin{pmatrix} C_{31} & C_{32} & 0 \end{pmatrix} \tag{4.36}$$

$$= \begin{pmatrix} A_{11} - A_{13}C_{33}^{-1}C_{31} & A_{12} - A_{13}C_{33}^{-1}C_{32} & A_{14} \\ A_{21} - A_{23}C_{33}^{-1}C_{31} & A_{22} - A_{23}C_{33}^{-1}C_{32} & A_{24} \\ C_{41} & C_{42} & 0 \end{pmatrix}. \tag{4.37}$$

Again, the operators R and P come from equations (4.30) to (4.32). Note that this reduction step is exact since C_{33} is a diagonal matrix. However, we have not eliminated the zero diagonal values after the first reduction step. Next, we eliminate the saturation block with the following C-F splitting:

$$S^1 = \left(\begin{array}{ccc} S_{11} & S_{12} & A_{14} \\ S_{21} & S_{22} & A_{24} \\ C_{41} & C_{42} & 0 \end{array}\right) \begin{array}{c} C \\ F \\ C \end{array} \tag{4.38}$$

The Schur complement (also the coarse grid) for the second level of multigrid reduction reads

$$S^2 = RS^1P = \begin{pmatrix} S_{11} & A_{14} \\ C_{41} & 0 \end{pmatrix} - \begin{pmatrix} S_{12} \\ C_{42} \end{pmatrix} \tilde{S}_{22}^{-1} \begin{pmatrix} S_{21} & A_{24} \end{pmatrix} \tag{4.39}$$

$$= \begin{pmatrix} S_{11} - S_{12}\tilde{S}_{22}^{-1}S_{21} & A_{14} - S_{12}\tilde{S}_{22}^{-1}A_{24} \\ C_{41} - C_{42}\tilde{S}_{22}^{-1}S_{21} & -C_{42}\tilde{S}_{22}^{-1}A_{24} \end{pmatrix}, \tag{4.40}$$

where $\tilde{S}_{22}^{-1}$ is some approximation of S_{22}^{-1} to compute R and P from (equation (4.30)). In the F-relaxation step, the action of the saturation block S_{22}^{-1} is achieved by one V-cycle of AMG. In the equation above, the presence of the constraints in the matrix S^2 makes it non-elliptic, and therefore, we cannot solve it using AMG, although it no longer has zeros on the diagonal. The final reduction step is employed to eliminate these constraints by putting them as F-points:

$$S^2 = \left(\begin{array}{cc} S_{11}^2 & S_{12}^2 \\ S_{21}^2 & S_{22}^2 \end{array}\right) \begin{array}{c} C \\ F \end{array} \tag{4.41}$$

$$S^3 = RS^2P = S_{11}^2 - S_{12}^2(\tilde{S}_{22}^2)^{-1}S_{21}^2 \tag{4.42}$$

The Schur complement at the last level S^3 can be solved using AMG. A schematic representation of the multi-level reduction approach is illustrated in figure 4.1. At

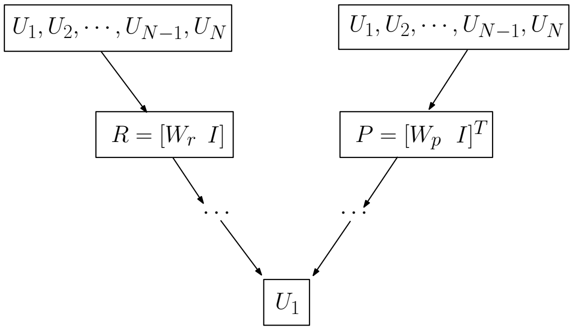

$U_1, U_2, \cdots, U_{N-1}, U_N$
$U_1, U_2, \cdots, U_{N-1}, U_N$
$R = [W_r \;\; I]$
$P = [W_p \;\; I]^T$
$\cdots$
$\cdots$
U_1

Figure 4.1: Multigrid reduction V-cycle

each level, we reduce one or more variables until we obtain the variable associated with the elliptic operator, which is the pressure variable in our case. This means that in figure 4.1, $U_1 \equiv P_l$.

4.6 Numerical Experiments

In this section, we perform numerical experiments to show the efficiency of the multigrid reduction approach (herein referred as hypreMGR) in solving the linear equations arising in various flow scenarios. We implement hypreMGR as a separate solver and preconditioner in Hypre [48, 49]. For all AMG solve steps, we use BoomerAMG [52], also included in Hypre. The two-phase, two-component flow using the NCP approach is implemented in Amanzi, the computational engine of teh Advanced Simulation Capability for Environmental Management (ASCEM) project [5]. Even though Amanzi was first designed for simulation of subsurface flow and reactive transport, its modular framework and concept of process kernels [35] allow new physics to be added relatively easily for other applications. The simulator employed in this work is one such example. Amanzi works on a variety of platforms, from laptops to supercomputers. It also leverages several popular packages for mesh infrastructure and solvers through a unified input file. Again, due to the presence of zeros on the diagonal of the generalized Jacobian matrix A, we cannot use current AMG solvers such as Hypre's BoomerAMG and Trillinos ML for our problem. Even when we eliminate the zeros on the diagonal of A to obtain a smaller system, using Hypre's BoomerAMG and Trilinos ML for system as a preconditioner, GMRES still fails to converge within 400 iterations. We also exclude ILUt variants (also implemented in Euclid and Trillinos) as they are not robust for the problems considered in this work. Thus, in all of our experiments we compare hypreMGR with the incomplete factorization (ILU) method from Euclid, which is also a part of Hypre. ILU(k) is used sequentially for all the examples. We experiment with different levels of fill k and report the results for the minimum k that is sufficient for GMRES to converge within 400 iterations throughout the simulation in each test case. GMRES is provided within Amanzi. For simplicity, we employ structured Cartesian grids for the test cases, but we can also use unstructured K-orthogonal grids. For parallel results, the test cases are run on Syrah, a Cray system with 5,184 Intel Xeon E5-2670 cores at the Lawrence Livermore National Laboratory Computing Center. Amanzi and other libraries are compiled with OpenMPI 1.6.5 and gcc-4.9.2. The total time is measured in seconds.

This section has four parts. In the first part, we show the results for an unsaturated flow problem with no phase appearance/disappearance. In the second part, we report the results for the saturated flow problem in which the gas phase appears by injection and then disappears after the injection is stopped. These two test cases were originally presented in the MoMaS gas benchmark project [19]. The third example includes a three-dimensional problem with parameters generally used in reservoir simulation. In the last part, we examine the scalability of the multigrid reduction approach.

Unless specified otherwise, for all of the simulations presented here, the convergence tolerance for semi-smooth Newton's method is $||F(x)|| \leq 10^{-5}$, and the linear tolerance for GMRES is $||J\delta u_k - F(u_k)|| \leq 10^{-12}||F(u_k)||$, which is the default in Amanzi. For AMG solves, we use the default parameters in BoomerAMG. The coarsening strategy is the parallel Cleary-Luby-Jones-Plassman (CLJP) coarsening [32]. The interpolation method is the classical interpolation defined in [80], and the smoother is the forward hybrid Gauss-Seidel/SOR scheme. The number of V-cycle steps is set to 1.

4.6.1 Unsaturated flow

This test shows a two-dimensional case in which the water and gas system is initially out of equilibrium and then evolves towards equilibrium. There is no flow in and out of the domain, and there is no phase appearance/disappearance. Figure 4.2 illustrates the porous domain, and the detailed set up of the experiment is as follows. For boundary conditions, we impose no flow condition on the boundary of the whole domain. Denoting $\psi^w = \rho_l^w \boldsymbol{K} \lambda_l \nabla P_l - j_l^h$ and $\psi^h = \rho_l^h \boldsymbol{K} \lambda_l \nabla P_l + \rho_g^h \boldsymbol{K} \lambda_g \nabla P_g + j_l^h$, we have $\psi^k \cdot \nu = 0, \ \ k \in \{w, h\}$ on Γ. Initial conditions are uniformly constant on each sub-domain Ω_1 and Ω_2:

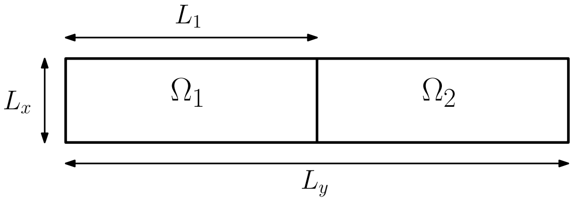


Figure 4.2: Porous domain Ω with two sub-domains Ω_1 and Ω_2.

- $P_l = P_{l,1}$ and $P_g = P_{g,1}$ on Ω_1.
- $P_l = P_{l,2} = P_{l,1}$ and $P_g = P_{g,2} \neq P_{g,1}$ on Ω_2.

For capillary pressure, we use the Van Genuchten model with $P_r = 2 \times 10^6$ Pa, $n = 1.54$, $S_{lr} = 0.01$, and $S_{gr} = 0$. The rest of the parameter values are shown in tables 4.1 and 4.2. We run 5 time steps of size $dT = 10$ seconds. The results are summarized in table 4.3. NS denotes the number of nonlinear iterations, LS the number of linear iterations, and LS/NS the average number of linear iterations per nonlinear iteration. In this experiment, because there is no phase disappearance/appearance, the diagonal of the Jacobian does not have any zeros, and ILU(0) can be used as a preconditioner. With respect to hypreMGR, we apply two levels of reduction for this problem, one for the constraints with nonzero diagonal values and one for the saturation block. The approximations for the restriction and interpolation operators from equation (4.31) are used in this case. As the liquid saturation does not go to zero, the effect of capillary pressure is small and that makes the system more advection-dominated. Thus, we do not need use AMG to solve for the saturation correction in the F-relaxation step. Here, we found that using three Gauss-Seidel smoothing steps is sufficient except for the finest grid where AMG with a two-level V(3,3)-cycle is used. For the coarse grid, a single AMG V(2,2)-cycle is

L_1	0.5 m
L_x	1 m
L_y	0.1 m
$P_{l,1}$	10^6 Pa
$P_{g,1}$	1.5×10^6 Pa
$S_{l,1}$	96.2 %
$P_{l,2}$	10^6 Pa
$P_{g,2}$	2.5×10^6 Pa
$S_{l,2}$	84.2 %

Table 4.1: Initial conditions

$\boldsymbol{K}$	$1 \times 10^{-16}\ m^2$
ϕ	0.3
D_l^h	$3 \times 10^{-9}\ m^2/s$
μ_l	1×10^{-9} Pa s
μ_g	9×10^{-6} Pa s
H	7.65×10^{-6} mol/Pa/m^3
M^h	2×10^{-3} kg/mol
M^w	1×10^{-2} kg/mol
ρ_w^{std}	10^3 kg/m^3

Table 4.2: Parameter Values

Mesh size	ILU(0)			hypreMGR		
	Time (s)	LS	LS/NS	Time (s)	LS	LS/NS
200 × 10	11.5	555	50.5	10.8	445	40.5
400 × 20	97.3	1283	98.7	42.2	458	35.2
800 × 40	757.4	2479	190.7	180.3	557	42.8
1600 × 80	5666	4321	332.4	801.8	569	43.8

Table 4.3: Performance of hypreMGR for different mesh sizes

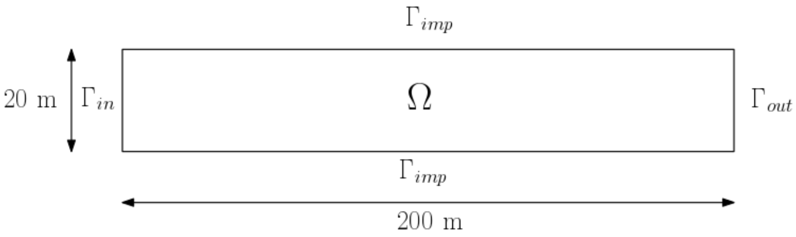


Figure 4.3: Core domain for the gas infiltration example.

applied for all the mesh sizes. Table 4.3 indicates that our new algorithm is more efficient both in terms of run time and number of iterations. It also exhibits near optimal scaling with respect to mesh size. hypreMGR is faster across all the meshes both in terms of the run time and average number of linear iterations. For the mesh size of 800 × 40, hypreMGR is twice faster in terms of run time, and takes less than a fifth of the number of iterations of ILU(0). For the largest mesh of 1600 × 80, ILU(0) is very inefficient, and hypreMGR is about 7 times faster than ILU(0) in terms of run time and number of iterations.

4.6.2 Saturated flow with phase appearance

This test is devoted to describing gas phase appearance produced by injecting pure hydrogen in a two-dimensional homogeneous porous domain Ω, which was initially 100% saturated by pure water. The domain is a rectangle of size 200m × 20m with three types of boundaries : Γ_{in} on the left side is the inflow boundary; Γ_{out} on the right side is the outflow boundary; and Γ_{imp} at the top and bottom is the impervious boundary (see figure 4.3). There is no source terms inside the boundary, and the boundary conditions are as follows

- No flux on Γ_{imp}

$$\psi^w \cdot \nu = 0 \text{ and } \psi^h \cdot \nu = 0 \tag{4.43}$$

- Injection of hydrogen on the inlet Γ_{in}

$$\psi^w \cdot \nu = 0 \text{ and } \psi^h \cdot \nu = 5.57 \times 10^{-6} \;\; kg/m^2/year \tag{4.44}$$

- Fixed liquid saturation and pressure on the outlet

$$P_l = 10^6 \;\; Pa, \quad S_l = 1, \quad \rho_l^h = 0 \tag{4.45}$$

Initial conditions are uniform throughout the domain, corresponding to a stationary state of saturated liquid and no hydrogen injection

$$P_l = 10^6 \;\; Pa, \quad S_l = 1, \quad \rho_l^h = 0 \tag{4.46}$$

The rest of the physical parameters are given in [19].

For the capillary pressure model, we experimented two scenarios: (1) power laws for relative permeabilities as in equation (4.11) in conjunction with the linear capillary pressure model, and (2) Van Genuchten for both relative permeabilities and capillary pressure model. In the first case, the entry pressure is $P_r = 2 \times 10^6$ Pa, and in the second case, we use $P_r = 2 \times 10^6$ Pa, $n = 1.49$. In both cases, the residual saturations are $S_{gr} = 0$ and $S_{lr} = 0.4$. We run the simulation for 100 time steps of fixed size $dT = 5000$ years. Figure 4.4 shows the infiltration of hydrogen after 5×10^5 years for the second scenario, and the performance of the preconditioners is reported in table 4.4. As can be seen in figure 4.4, the left side of

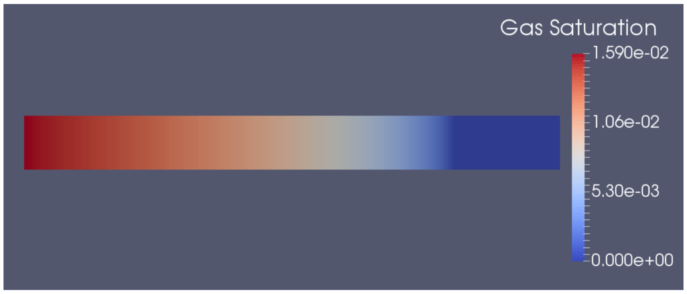


Figure 4.4: Gas infiltration after 100 time steps

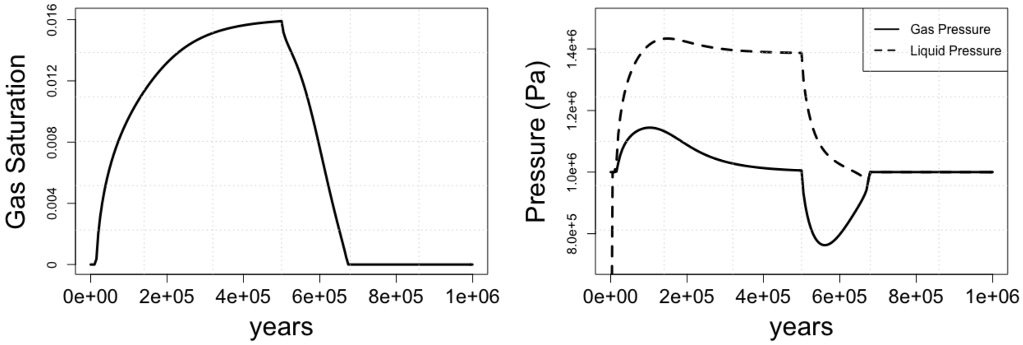


Figure 4.5: Gas saturation and pressure profiles at the leftmost cell over time for the saturated flow case.

the core is infiltrated with hydrogen, while the right side is still fully saturated with water. We also plot the gas saturation and the pressures in the first cell over time in figure 4.5. Although we do not have the exact numbers for comparison, a visual inspection indicates that our simulation results match well with those in [19,62,65].

Regarding the setup of hypreMGR in this case, we use three levels of reduction with the restriction and interpolation operators in equation (4.32). For the first level which we need to eliminate the constraints with non-zero diagonal values, a single Jacobi iteration used for the F-relaxation. For the subsequent levels, we apply a singe AMG V(1,1)-cycle for the A_{ff} solve. The coarse grid correction is also solved with one AMG V(2,2)-cycle. With regard to ILU(k), we experiment with different levels of fill and find that ILU(0), ILU(1), etc. would fail to converge for some time step, and ILU(5) is needed for convergence throughout the simulation. For the nonlinear Van Genuchten model, the new approach requires about 34% fewer number of iterations and about 17% less time than ILU(5) for mesh size of 200×10, as shown in table 4.4. The advantage of this approach is much clearer as the problem gets larger (see table 4.5). In this case, hypreMGR takes fewer than half the number of iterations of ILU(5) and requires 40% less time. Even though the average number of iterations does grow in the case of hypreMGR, it is much less than the rate of ILU(5). Also, we suspect that the mesh may not be large enough for the method to show mesh independence. Similarly, the new approach outperforms ILU(5) in both number of iterations and run time for the linear model.

Methods	Linear			Van Genuchten		
	Time (s)	LS	LS/NS	Time (s)	LS	LS/NS
ILU(5)	505	22595	28.9	522	24474	34.7
hypreMGR	485	16554	21.2	433	15234	21.6

Table 4.4: Performance of Euclid ILU v.s. hypreMGR for mesh size 200 × 10

Methods	Linear			Van Genuchten		
	Time (s)	LS	LS/NS	Time (s)	LS	LS/NS
ILU(5)	3442	42835	60.3	3467	43949	64.6
hypreMGR	2291	20408	28.7	2096	19122	28.3

Table 4.5: Performance of Euclid ILU v.s. hypreMGR for mesh size 400 × 20

Mesh size	ILU(5)			hypreMGR		
	Time (s)	LS	LS/NS	Time (s)	LS	LS/NS
20^3	194.0	690	25.6	121.1	267	9.9
40^3	2715.1	1470	49.0	1381.2	397	13.2

Table 4.6: Performance of hypreMGR for different mesh sizes

4.6.3 Three-dimensional Case with Phase Transition

The domain is a box of dimensions 100m × 100m × 100m. We use a homogeneous permeability field of $\boldsymbol{K} = 10^{-14}$ m^2, which is typical for fresh sandstone (see [11]) that is prevalent in reservoir simulation. The domain is saturated with water, and pure hydrogen is injected into the domain through the boundary of a corner at the bottom. The outlet is set at the opposite corner. The injection rate is 3×5.57 kg/m^2/year . We run 1 time step of size $dT = 1.825$ days. For the relative permeabilities and capillary pressure models, we use the Van Genuchten model with the same parameters as the example presented in section 6.3. The results are shown in table 4.6. Since this is a case with phase appearance/disappearance, we use the same setup as in the second example (section 6.2) for hypreMGR. From table 4.6, the new approach is about 40% faster in terms of run time, and takes fewer than half the number of iterations of ILU(5) for the mesh size of 20^3. For the larger mesh size of 40^3, it is twice faster in terms of run time, and it takes four times fewer the number of iterations of ILU(5). The result indicates that the advantage of hypreMGR clearer as the problem gets larger. Although similar to the result in the previous example, there is an increase in the number of iterations for hypreMGR in the case of the larger mesh, but again the problem is not large enough for us to see the mesh independence result. In fact, we show that this is exactly the case in the next section. And even though the results are not presented here, we note that hypreMGR also outperforms ILU(5) both in run time and number of iterations for the linear model of capillary pressure.

4.6.4 Scaling Results

For the scalability study, we use the same problem setup as in the three-dimensional example in section 6.3. The only difference is in the mesh size. For a strong scaling study, we fix the mesh at 80^3 (about 1.5 million unknowns) and run the simulation on 8 to 128 cores, each time doubling the number of cores. For weak scaling, we start with a mesh of 40^3 and then refine the mesh in all directions up to 320^3, so the largest problem has about 100 million unknowns. We run the problem with 2, 16, 128, and 1024 cores, respectively. The initial time step is $dT = 0.125$ day, and the final time of the simulation is 10 days, except for the case of the largest mesh, which we stop the simulation at 3 days, as we reach the memory limit of the machine.

The results in figure 4.6 shows that hypreMGR achieves promising results, scaling well up to 64 cores, although it is not quite optimal. From 64 to 128 cores, however, there run time actually increases. This is due to the problem size on each

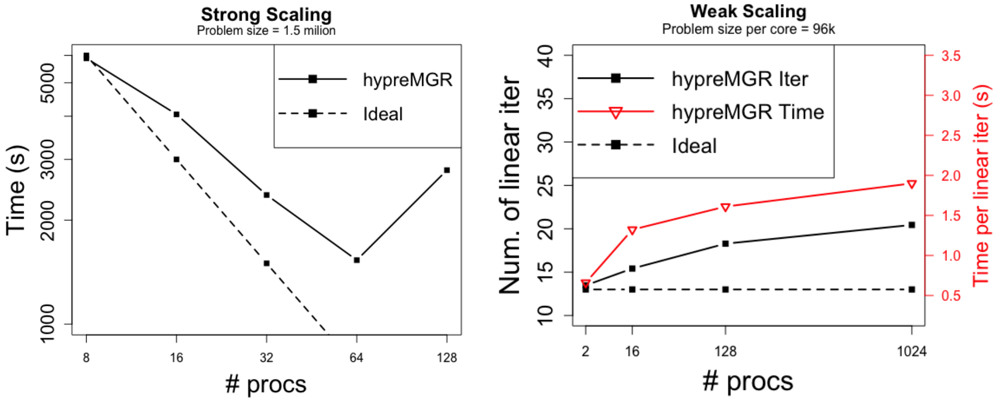


Figure 4.6: Scaling results for MGR.

processor getting small (about 12,000 unknowns for 128 cores), and as a consequence, the computation to communication ratio decreases, and that makes the method less efficient. For weak scaling, the performance of hypreMGR is independent of the mesh size. As the problem size gets larger (8 times for each refinement level), the number of linear iterations per nonlinear iterations does not grow significantly, about 14, 19, and 12 percent for 16, 128, and 1024 cores, respectively. The average number of linear iterations also seems to approach a limit, which demonstrates optimal multigrid performance. Regarding run time, we measure both the setup phase and the solve phase of the algorithm. Since the setup phase requires expensive matrix-matrix multiplications, the total time needed to solve a linear system grows a little faster than the number of iterations. Yet, hypreMGR still achieves near optimal scalability. Figure 4.7 focuses on the time of the linear solve, splitting into the setup and solution phases. It is clear that solve phase achieves optimal scalability as the time needed to iterate to convergence stays nearly constant for mesh sizes 80^3, 160^3, and 320^3. In contrast, the setup phase, which includes constructing R and

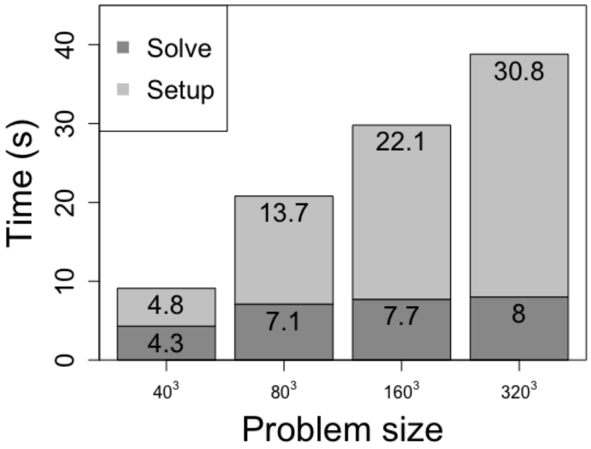


Figure 4.7: CPU time breakdown for linear solve.

P, computing the coarse grids using the matrix-matrix product RAP, and all the AMG setup for the coarse grid as well as the F-relaxation, does not scale very well. This is likely an implementation problem, and it can be improved in the future.

In terms of memory storage, like AMG methods, the MGR approach requires the storage of the restriction, interpolation, and coarse grid operators at every level. However, in addition to these operators, the MGR approach also needs to store the A_{ff} matrices at the levels which scalar AMG is used for the F-relaxation step. When aggressive coarsening is performed in standard AMG methods, the size of the coarse grid can be significantly reduced after the first level. With MGR, the size of the matrix is reduced by at most a third, since the reduction is dictated by the block structure of the system, rather than the heuristics used in AMG.

4.7 Conclusion

We have presented a preconditioning strategy for solving the linear systems that arise from the solution of multiphase multicomponent porous media flow with phase transitions. To account for the phase transitions, the problem is formulated as a nonlinear complementarity problem, and solved using the semi-smooth Newton technique.

The proposed preconditioner is based on the multigrid reduction technique, which generalizes traditional two-stage preconditioners in a natural multigrid framework. In this work, we extend a previously developed two-grid strategy to a multilevel reduction strategy that accounts for the transitions in the phases of the primary variables. We have demonstrated the performance of the preconditioner on classic benchmark problems presented in the literature, and show the parallel efficiency of the linear solver on large-scale problems. The numerical results indicate optimal scalability and robust performance of the MGR preconditioner, which is important for real-field simulations.

We observed that depending on the properties of the capillary pressure model used, a different solver could be used for the F-relaxation phase of the preconditioner. When the model is convection-dominated, a simple relaxation scheme is sufficient for F-relaxation. However, when the model is diffusion-dominated, relaxation alone is not sufficient, and a more robust solver is required for F-relaxation. In our experiments, we used AMG for such problems. However, this may be excessive and in some cases, inefficient. For applications with phase transitions, the fronts along which the transitions occur can be small compared to the entire domain. As a result, using the same strategy for F-relaxation at these intermediate solves can be inefficient since communication dominates computation at this point. Allowing different strategies to be employed, as dictated by the physics, can be a more efficient strategy. For example, using a single-level relaxation strategy instead of a multilevel (v-cycle) technique could be more appropriate. We are exploring this idea, in addition to aggressive coarsening strategies to improve parallel efficiency.

Future applications of interest for the MGR solver include applications with multiple phases, poromechanics, and applications with fractures and thermal properties. The MGR framework is general enough to handle these applications as a "black-box" solver, and can also serve as a basis for building good physics-based preconditioners. However, more work may be required to improve solver performance for these complex applications. We are exploring new strategies for building interpolation and restriction operators so that the final coarse grid system is a good approximation to a pressure system (or has elliptic M-matrix properties) that is amenable to solution by AMG. We are also considering incorporating structure information within a semi-structured framework to develop a robust solver that can effectively handle grid anisotropy for complex geometries.

Chapter 5: Semi-smooth Newton Methods for Compositional Two-phase Flow in Porous Media with Phase Transitions

5.1 Introduction

In chapter 4, we explored an NCP formulation of a compositional two-phase flow model and developed a new preconditioner for the linear systems arising in solving problems with phase transitions. The robustness and scalability of this preconditioner allows us to run complex simulations on larger domains for longer time periods. These simulations, however, also reveal the lack of robustness and efficiency of the semi-smooth Newton method (algorithm 4 in chapter 4) in resolving phase transitions when it is applied to highly nonlinear problems with heterogeneous media. Thus, the goal of this chapter is to develop a new method for the NCP formulation that can overcome these limitations.

Phase transitions have posed a major challenge for multi-phase, multi-component models since the 1980s. If not handled correctly, they can cause numerical oscillations in solutions of these models, making such solutions physically inconsistent and unusable. There have been many attempts to address the problems with phase transitions and determine the correct local thermodynamic state for compositional multiphase flow. In general, most of these can be classified into two common classes of methods: flash calculation [2, 25, 26, 34, 64, 101] and PVS [51, 98]. Flash calculation computes the local thermodynamic state from the overall mass of the individual components. While this method is stable with regard to determining the thermodynamic state, it tends to be inefficient because it requires solution of a large nonlinear system of equations at each time step (in addition to solution of the linearized systems) to recover all the thermodynamic quantities of interest. The second class, PVS, involves adapting the primary variables to the thermodynamic constraints locally. The idea is that whenever phase transitions occur, physical variables that are physically inconsistent (indicated by negative saturation, for example) are switched to well-defined quantities. The governing equations related to those variables are also modified accordingly. Although this approach is locally more efficient than flash calculations, it suffers from irregular convergence behavior in the nonlinear solve, which is typically addressed by substantial reduction in time step size [31]. This feature is not desirable for simulations over a long period of time, usually encountered in groundwater remediation or transport of nuclides in a nuclear waste repository. In addition to flash calculations and PVS, there are other formulations to handle phase transitions such as negative saturation [1], and introduction of persistent primary variables [18, 63, 65].

Recently, a new approach has been developed for handling the phase transitions by formulating the system of equations as a nonlinear complementarity problem (NCP) [16, 59, 62]. In contrast to PVS, NCP has the advantage that the set of primary variables is consistent throughout the simulation, and no primary variable switching is needed. NCP requires a complementary function, referred to as a C-function, employed to rewrite the inequality constraints for the thermodynamic state as a non-smooth equation, which requires a *semi-smooth* Newton method [3,71,73] to solve. Most of the previous work in multiphase flow using the NCP approach employs the minimum function as the C-function due to its simplicity for implementation and the fact that it is piecewise linear with respect to the arguments. Even though the semi-smooth Newton method applied to the NCP using the minimum function as a C-function is observed to have quadratic convergence for simple problems in porous media (see [16]), it exhibits poor convergence and even diverges for many problems considered in this chapter. An alternative to the minimum function is the Fischer-Burmeister function, which has recently been employed as the C-function for NCP formulation of incompressible two-phase flow in [99]. As we will show, this choice of C-function can help mitigate the lack of robustness observed in using the Newton-min algorithm for NCP formulation of compositional two-phase flow with phase transitions. We then draw on this experience and develop a new method for the nonlinear solve based on a smooth version of the Fischer-Burmeister function. Our method can be considered a variant of the Jacobian smoothing method summarized in [46]. Compared to the non-smooth approaches that use the minimum and the Fischer-Burmeister functions, our new method is more robust and efficient for problems with highly heterogeneous media, and it also scales optimally with problem size.

We consider a two-phase, two-component system with phase transitions as our model problem. We describe this model in detail in section 5.2, and in section 5.3, we describe the NCP formulation for it. We briefly review the semi-smooth Newton framework and introduce our new algorithm in section 5.4. In section 5.5, several numerical tests are presented that demonstrate the robustness and scalability of the new algorithm. Some concluding remarks as well as future work are presented in section 5.6.

5.2 Problem Statement

5.2.1 Governing Equations

Here, we consider the same simplified two-phase two-component model with phase transitions presented in section 4.2. To make it easier to follow, we restate the complete sytem of equations below. For each component, conservation of mass leads to

$$\phi \frac{\partial(\rho_l^w S_l)}{\partial t} + \nabla \cdot (\rho_l^w \boldsymbol{q}_l - j_l^h) = 0, \tag{5.1}$$

$$\phi \frac{\partial(\rho_l^h S_l + \rho_g^h S_g)}{\partial t} + \nabla \cdot (\rho_l^h \boldsymbol{q}_l + \rho_g^h \boldsymbol{q}_g + j_l^h) = 0, \tag{5.2}$$

where the subscripts l, g denote the liquid and gas phases, and the superscripts w, h denote the water and hydrogen components, respectively. The porosity of the medium is denoted ϕ, S_α, $\boldsymbol{q}_\alpha$ are the saturation and velocity of phase α, respectively; ρ_l^h is the dissolved hydrogen mass concentration in the liquid phase; and j_l^h is the diffusion flux of hydrogen in the liquid phase. The Darcy velocity $\boldsymbol{q}_\alpha$ follows the Darcy-Muskat law

$$\boldsymbol{q}_\alpha = -\boldsymbol{K}\lambda_\alpha \nabla(P_\alpha - \rho_\alpha \boldsymbol{g}), \quad \alpha = l, g, \tag{5.3}$$

where $\boldsymbol{K}$ is the absolute permeability, λ_α, P_α, and ρ_α are the mobility, pressure, and density of phase α, and $\boldsymbol{g}$ is the gravitational acceleration. The mobility λ_α of phase α is defined as the ratio between the phase relative permeability $k_{r\alpha}$ and the phase viscosity μ_α: $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$. Using Fick's law, the diffusion flux of hydrogen in liquid j_l^h in equations (5.1) and (5.2) can be expressed as

$$j_l^h = -\phi S_l D_l^h \nabla \rho_l^h, \tag{5.4}$$

where D_l^h is the hydrogen molecular diffusion coefficient in liquid. Since we assume incompressibility of the liquid phase, the mass density of the water component in the liquid phase is constant, i.e. $\rho_l^w = \rho_w^{std}$. To capture capillarity effects, the jump in the pressure at the interface of the two phases is modeled by the relation

$$P_g = P_l + P_c(S_l) \tag{5.5}$$

where P_c is the capillary pressure. Additionally, we have the constraints

$$S_l + S_g = 1. \tag{5.6}$$

To close the model, we also need a set of equations for the thermodynamic equilibrium when the gas phase is present, i.e. how much hydrogen can dissolve into the liquid phase at a certain pressure. Assuming low solubility of hydrogen in the liquid phase, Henry's law can be used to connect the gas pressure P_g and the dissolved hydrogen mass concentration in liquid ρ_l^h:

$$\rho_l^h = C_h P_g, \tag{5.7}$$

where $C_h = HM^h = \rho_w^{std} M^h / M^w$, H is the Henry's law constant, and M^i, $i \in \{w, h\}$, is the molar mass of the i-th component. Since we neglect water vapor, we can apply the ideal gas law for the gas phase. This leads to the relation

$$\rho_g^h = \rho_g = C_v P_g, \tag{5.8}$$

where C_v is a constant and $C_v = M^h/(RT)$; T is the temperature and R the ideal gas constant.

5.2.2 Relative Permeabilities and Capillary Pressure

In this chapter, we employ the nonlinear Van Genuchten [91] model for relative permeabilities and capillary pressure:

$$k_{rl} = \sqrt{S_{le}}\Big(1 - \big(1 - S_{le}^{1/m}\big)^m\Big)^2, \quad k_{rg} = \sqrt{1 - S_{le}}\Big(1 - S_{le}^{1/m}\Big)^{2m}, \tag{5.9}$$

$$P_c = P_r\Big(S_{le}^{-1/m} - 1\Big)^{1/n}, \tag{5.10}$$

$$S_{le} = \frac{1 - S_l}{1 - S_{lr} - S_{gr}}, \quad m = 1 - \frac{1}{n}, \tag{5.11}$$

where P_r is the entry pressure. Notice that the function $P_c(S_l)$ in the Van Genuchten model is only defined for $S_l \in [S_{lr}, 1 - S_{gr}]$ and P_c' is unbounded near S_{lr} and $1 - S_{gr}$. Thus, it is necessary to modify the model to limit the growth of P_c' and extend it for $S_l \in \mathbb{R}$, since the value of S_l can become larger than $1 - S_{gr}$ or less than S_{lr} during the nonlinear iteration. We use a regularization as presented in [62] with parameter $\epsilon = 10^{-5}$. In this regularization, the capillary pressure is computed by a linear extrapolation from the regularization points $S_{lr} + \mathcal{O}(\epsilon)$ and $1 - S_{gr} - \mathcal{O}(\epsilon)$ with the slopes $P_c'(S_{lr} + \mathcal{O}(\epsilon))$ and $P_c'(1 - S_{gr} - \mathcal{O}(\epsilon))$, respectively.

5.2.3 Primary Variables

There are many ways to choose a set of primary variables, depending on the problem formulation and applications. In our model example, a convenient choice is the liquid pressure, liquid saturation, and the concentration of hydrogen in the liquid phase. We then have our solution vector $u = \{P_l, S_l, \rho_l^h\}$. Unlike in other methods such as primary variable switching, for NCP, the choice of primary variables is fixed throughout the simulation.

5.3 Nonlinear Complementarity Problem

In its simplest form, a nonlinear complementarity problem with respect to a smooth function $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ is to find a vector $x \in \mathbb{R}^n$ such that

$$x \geq 0, \quad f(x) \geq 0, \quad x^T f(x) = 0, \tag{5.12}$$

A slightly more general form of the last equation in equation (5.12) reads

$$g(x)^T f(x) = 0,$$

where $g : \mathbb{R}^n \to \mathbb{R}^n$ is another smooth function. As we have mentioned in section 5.2, for the solution to be valid, the pressure, saturation, and hydrogen concentration in the liquid phase must satisfy the constraints in equations (5.6) and (5.7). These conditions can be reformulated as an NCP as follows:

$$1 - S_l \geq 0, \quad C_h P_g - \rho_l^h \geq 0, \quad (1 - S_l)(C_h P_g - \rho_l^h) = 0. \tag{5.13}$$

As we have observed, methods based on NCP require a complementarity function, also called C-function, $\Phi(a,b) : \mathbb{R}^2 \to \mathbb{R}$, which satisfies

$$\Phi(a,b) = 0 \iff a \geq 0, \quad b \geq 0, \quad ab = 0. \tag{5.14}$$

Solving the NCP problem in equation (5.13) is equivalent to solving $\Phi(a,b) = 0$ with functions $a(u) = 1 - S_l$ and $b(u) = C_h P_g - \rho_l^h$. There are many examples of C-functions. In this work, we focus on two popular choices

$$\Phi_{\min}(a,b) = \min(a,b) \tag{5.15}$$

$$\Phi_{FB}(a,b) = \sqrt{a^2 + b^2} - (a+b) \quad \text{(Fischer-Burmeister)} \tag{5.16}$$

The minimum function is convenient because it is piecewise linear with respect to the variables a and b, which simplifies the computation of the Jacobian in each nonlinear iteration. When the gas phase is not present, equation (5.15) reduces to $1 - S_l = 0$. When the gas phase appears, $1 - S_l > 0$ and the constraint equation is governed by Henry's law equation (5.7). However, compared to the Fischer-Burmeister function, the minimum function is less useful with respect to globalization with line search strategies.

First, as shown and discussed in [13–15], global semi-smooth Newton methods may diverge even for linear C-functions if the starting point is not close enough to a solution. Second, in global semi-smooth Newton approaches, at each step of the iteration, one needs to find an appropriate step size α_k for the solution update $x_{k+1} = x_k + \alpha_k \mathbf{p}_k$, where $\mathbf{p}_k$ is a search direction. This can be accomplished by introducing a *merit function*[1]. Since the step size is usually chosen based on the derivative of the

[1] For a precise definition and discussion of merit functions, we refer to [66].

merit function, it is desirable for the merit function to be continuously differentiable. The merit function associated with the minimum function $\Psi_{\min} = \|\Phi_{\min}(a,b)\|^2$, however, does not satisfy this condition. In contrast, the Fischer-Burmeister merit function $\Psi_{FB} = \|\Phi_{FB}(a,b)\|^2$ is continuously differentiable.

5.4 Solution Algorithm

We consider solving the coupled system consisting of equations (5.1), (5.2) and (5.13) fully implicitly. We use a cell-centered finite volume method for spatial discretization, as it is a natural way to preserve the mass conservation property of the balance equations (5.1) and (5.2). In addition, it can deal with the case of discontinuous permeability coefficients, and it is relatively straightforward to implement. For the time domain, we employ the backward Euler method to avoid a CFL stability restriction on the time step. Because this method is unconditionally stable, it also allows us to experiment with variable time stepping, which can significantly reduce execution time.

5.4.1 Semi-smooth Newton Method

Let us introduce a discrete version of the C-function equation (5.14) with respect to the solution vector u introduced in section 5.2.3,

$$\Theta(u) = \begin{pmatrix} \Phi(1-(S_l)_1, (C_h P_g - \rho_l^h)_1) \\ \Phi(1-(S_l)_2, (C_h P_g - \rho_l^h)_2) \\ \dots \\ \Phi(1-(S_l)_N, (C_h P_g - \rho_l^h)_N) \end{pmatrix} \tag{5.17}$$

where N is the number of cells in the mesh. Thus, Θ is the discrete function of the reformulation of the constraints using a C-function. We want to solve the system $R(u) = 0$ where $R(u)$ is the residual function given by

$$R(u) = \begin{cases} H(u) & \text{(from the PDEs)} \\ \Theta(u) & \text{(from the constraints)} \end{cases} \tag{5.18}$$

A straightforward approach for solving nonlinear systems of equations is Newton's method, which requires solution of a linear system at each iteration k:

$$\frac{\partial R}{\partial u}\Big|_{u=u_k} \delta u = -R(u_k). \tag{5.19}$$

This method requires that the Jacobian $\partial R/\partial u$ be defined everywhere. In the NCP formulation, the constraints Θ are not differentiable when there is phase transition as the solution changes from satisfying $1 - S_l = 0$ to $C_h P_g - \rho_l^h = 0$. To address this, we will consider a semi-smooth Newton method, which is similar to Newton's method, except the derivative Θ' is replaced by a member of the *subdifferential* $\partial\Theta$ when Θ is not differentiable. Let $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a locally Lipschitz-continuous function and D_F be the set where F is differentiable; the B-subdifferential of F at x is defined as the set

$$\partial_B F(x) := \{G \in \mathbb{R}^{n\times n} : \exists\, x_k \in D_F \text{ with } x_k \to x, \nabla F(x_k) \to G\} \ .$$

Below is the algorithm for the general semi-smooth Newton method (see [46]), of which algorithm 4 in chapter 4 is a particular case.

Algorithm 6 General semi-Smooth Newton method.

```
while $k < max\_iter$ and $res > tol$ do
    (1) Given $u^0$, $k = 0$
    (2) Select an element $J_k \in \partial_B \Theta(u^k)$
    (3) Solve the system
    $\begin{pmatrix} H'(u^k) \\ J_k(u^k) \end{pmatrix} \triangle\, u^k = \begin{pmatrix} -H(u^k) \\ -\Theta(u^k) \end{pmatrix}$
    (4) Update $u^{k+1}$
    $u^{k+1} = u^k + \triangle u^k$
```

To compute J_k in the algorithm above, one can use an active set strategy [54]. In the context of multiphase flow with phase appearance and disappearance, the idea is to define the set of indices for the cells in which the gas phase is present (see [16,59]). Let $A^k := \{j : 1-(S_l)_j \geq (C_h P_g - \rho_l^h)_j\}$, $I^k := \{j : 1-(S_l)_j < (C_h P_g - \rho_l^h)_j\}$. Then for the minimum function, the jth row of J_k is equal to

$$\begin{cases} \dfrac{\partial}{\partial u} a(u)_j \ \text{ if } \ j \in I^k \\ \dfrac{\partial}{\partial u} b(u)_j \ \text{ if } \ j \in A^k \end{cases} \tag{5.20}$$

where $a(u)_j = 1 - (S_l)_j$ and $b(u)_j = (C_h P_g - \rho_l^h)_j$. For the Fischer-Burmeister function, we can compute the jth row of J_k as follows,

$$\begin{cases} \dfrac{1}{\sqrt{a(u)_j^2 + b(u)_j^2}}\Big(a(u)_j \dfrac{\partial}{\partial u} a(u)_j + b(u)_j \dfrac{\partial}{\partial u} b(u)_j\Big) - \\ \Big(\dfrac{\partial}{\partial u} a(u)_j + \dfrac{\partial}{\partial u} b(u)_j\Big) & \text{if} \quad a(u)_j^2 + b(u)_j^2 \neq 0 \\ (\alpha_i - 1)\dfrac{\partial}{\partial u} a(u)_j + (\beta_i - 1)\dfrac{\partial}{\partial u} b(u)_j & \text{otherwise} \end{cases} \tag{5.21}$$

where for all i such that $a(u)_j^2 + b(u)_j^2 = 0$, α_i and β_i are arbitrary nonnegative constants satisfying $\alpha_i^2 + \beta_i^2 = 1$. For a more complete treatment of semi-smooth Newton methods, we refer to [46].

5.4.2 Jacobian Smoothing Method

An alternative to the semi-smooth approach is to employ a smooth approximation to the non-smooth function Θ. Let $G : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}^n$ such that for any $\tau > 0$, $G(\cdot, \tau)$ is continuously differentiable on $\mathbb{R}^n$ and

$$\|\Theta(u) - G(u, \tau)\| \to 0, \quad \text{as} \quad \tau \to 0. \tag{5.22}$$

Then, given a sequence τ^k, $k = 0, 1, 2, ...$, we can solve the system in equation (5.18) inexactly using $G'(u^k, \tau^k)$ as an approximation to the generalized Jacobian $J_k = \partial_B \Theta(u^k)$. In this work, we explore a smooth approximation to the Fischer-Burmeister functions given by

$$G_{FB}(u, \tau) = \sqrt{a^2 + b^2 + 2\tau} - (a + b) \tag{5.23}$$

The complete algorithm is as follows:

Algorithm 7 Jacobian Smoothing Method.

```
while $k < max\_iter \ and \ res > tol$ do
    (1) Given $u^0$, $k = 0$, and $\tau^0$
    (2) Solve the system
    $\begin{pmatrix} H'(u^k) \\ G'(u^k, \tau^k) \end{pmatrix} \triangle u^k = \begin{pmatrix} -H(u^k) \\ -\Theta(u^k) \end{pmatrix}$
    (3) Update the smoothing parameter $\tau$
    $\tau^{k+1} = \beta \tau^k$ for $\beta \in (0, 1)$
    (4) Update $u^{k+1}$
    $u^{k+1} = u^k + \triangle u^k$
```

There also exist smooth approximations to the minimum function. In particular, we experimented with the Chen-Harker-Kanzow-Smale smoothing [24, 56],

$$G_{min}(u, \tau) = (a + b) - \sqrt{(a - b)^2 + 4\tau}. \tag{5.24}$$

However, in our experience, this smooth version of the minimum function does not improve the convergence of the semi-smooth Newton's method significantly, and we do not include the results here.

5.4.3 Linear System

Assuming that each physical variable is ordered lexicographically, each step of the nonlinear iteration (step 3 in algorithm 6 and step 2 in algorithm 7) requires solution of a large sparse, non-symmetric, indefinite linear system of the form

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}, \tag{5.25}$$

in which the matrices in the first two rows are the discretized version of the linearized operators from the PDEs, and the last row corresponds to the discrete derivative of the complementarity constraint equation introduced in equation (5.17). Iterative methods such as GMRES [84] are the only viable option to solve the system above, and preconditioning is critical for fast convergence. Here, all of our experiments use GMRES preconditioned with hypreMGR, the AMG preconditioner we discussed in chapter 4. Unlike ILU preconditioners in which one only needs to specify the level of fill (in ILU(k)) or the threshold tolerance (in ILU(t)), hypreMGR requires extra information regarding the block structure of the system and the order of reduction.

There exists a small but important difference in the structure of the matrices A created using the Jacobian smoothing method and the semi-smooth Newton methods. For the semi-smooth Newton methods with an active set strategy, the diagonal of the block A_{33} contains zeros for the cells that are devoid of the gas phase (see section 4.3.2 for the case with the minimum function). In contrast, for $\tau > 0$, the diagonal of the block A_{33} is guaranteed to be nonzero for the Jacobian smoothing method regardless of the existence of phase transitions. Thus, using the Jacobian smoothing method, we do not need to partition the block A_{33} as in equation (4.24). The last reduction step in equation (4.42) can be omitted as a consequence. This means that the Jacobian method requires only two levels of reduction for hypreMGR, as opposed to three levels in the case of the semi-smooth Newton approach presented in chapter 4. Using fewer reduction steps helps decrease both the number of GMRES iterations and execution time, as will become evident from the results presented in section 5.5.3.

5.5 Numerical Results

In this section, we perform numerical experiments for the semi-smooth Newton's approach using the minimum and the Fischer-Burmeister functions, and the Jacobian smoothing method with the smooth Fischer-Burmeister function for the NCP formulation. All of these methods are implemented in Amanzi. GMRES is also provided within Amanzi while hypreMGR is employed through HYPRE.

This section has three parts. In the first part, we show the results for two benchmark problems that aims to show the effectiveness of the NCP approach in handling phase appearance and disappearance. In the second part, we report the results for both two- and three-dimensional cases with highly heterogeneous media. In the last part, we perform a scalability study of the new algorithm.

For all the simulations presented here, the convergence tolerance for nonlinear step is $||F(x)|| \leq 10^{-6}$, and the linear tolerance for GMRES is $||J\delta u_k - F(u_k)|| \leq 10^{-12}||F(u_k)||$, which is the default in Amanzi. Depending on the performance of the nonlinear solver, a heuristic for choosing the time step is used: if the number of nonlinear steps (NS) required at a given time are less than 10, then the next time step dt_{next} is doubled, $dt_{next} = 2 \cdot dt$; if NS $\in [11, 15]$, then the time step is kept fixed,

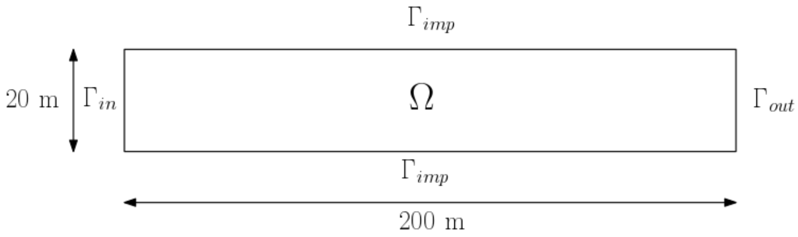


Figure 5.1: Core domain for the gas infiltration example.

$dt_{next} = dt$; and if NS is greater than 15, then the time step is halved $dt_{next} = dt/2$. The maximum number of nonlinear iteration is $max\_iter = 20$.

5.5.1 Benchmark problems

These tests are derived from the MoMaS benchmark project [19], which is designed to evaluate the effectiveness of different approaches for handling gas phase appearance and disappearance. Pure hydrogen is injected into a two-dimensional homogeneous porous domain Ω, which was initially 100% saturated with pure water. The domain is a rectangle of size 200m × 20m, and it is discretized only in the horizontal direction, leading to a quasi one-dimensional problem. There are three types of boundaries : Γ_{in} on the left side is the inflow boundary; Γ_{out} on the right side is the outflow boundary; and Γ_{imp} at the top and bottom is the impervious boundary (see figure 5.1). There are no source terms inside the domain, and denoting $\psi^w = \rho_l^w \boldsymbol{K}\lambda_l \nabla P_l - j_l^h$ and $\psi^h = \rho_l^h \boldsymbol{K}\lambda_l \nabla P_l + \rho_g^h \boldsymbol{K}\lambda_g \nabla P_g + j_l^h$, the boundary conditions are as follows

- No flux on Γ_{imp}

$$\psi^w \cdot \nu = 0 \text{ and } \psi^h \cdot \nu = 0 \tag{5.26}$$

- Injection of hydrogen on the inlet Γ_{in}

$$\psi^w \cdot \nu = 0 \text{ and } \psi^h \cdot \nu = 5.57 \times 10^{-6} \;\; kg/m^2/year \tag{5.27}$$

- Fixed liquid saturation and pressure on the outlet

$$P_l = 10^6 \;\; Pa, \quad S_l = 1, \quad \rho_l^h = 0 \tag{5.28}$$

Initial conditions are uniform throughout the domain, corresponding to a stationary state of saturated liquid and no hydrogen injection,

$$P_l = 10^6 \;\; Pa, \quad S_l = 1, \quad \rho_l^h = 0. \tag{5.29}$$

The values of the physical parameters are given in table 5.1.

Figure 5.2 shows the Van Genuchten capillary pressure curve for different values of the entry pressure P_r. These parameter values, along with others in the Van Genuchten model, depend on the porous material. For example, the MoMaS benchmark problem (test case 1 in [19]) uses $P_r = 2 \cdot 10^6$ for a very dense rock with extremely low permeability of $\boldsymbol{K} = 5 \cdot 10^{-20}$. In other applications including CO_2 sequestration and reservoir simulation, the material is much more permeable and $P_r = 2 \cdot 10^3$ would produce the capillary pressure curves typically used (see [30,67]). Other parameters for the Van Genuchten model are $S_{lr} = 0.4$, $S_{gr} = 0$, and $n = 1.49$. The smaller P_r is, the steeper the curve becomes near $S_l = 0$, and that also makes the problem more difficult to solve. The effect of capillary pressure on the solution is shown in figure 5.3, in which the gas saturation throughout the domain is plotted at 100,000 years for $P_r = 2 \cdot 10^6$ and $P_r = 2 \cdot 10^3$. For the MoMaS benchmark

$\boldsymbol{K}$	5×10^{-20} m^2
ϕ	0.15
D_l^h	3×10^{-9} m^2/s
μ_l	1×10^{-9} Pa s
μ_g	9×10^{-6} Pa s
H	7.65×10^{-6} mol/Pa/m^3
M^h	2×10^{-3} kg/mol
M^w	1×10^{-2} kg/mol
ρ_l^w	10^3 kg/m^3

Table 5.1: Parameter Values

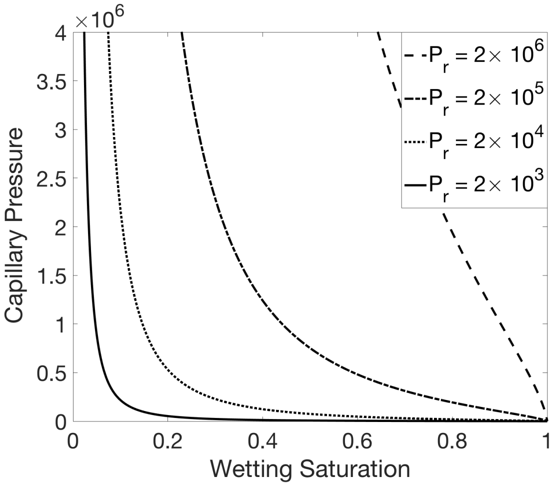


Figure 5.2: Capillary pressure curves for different entry pressure P_r.

case with $P_r = 2 \cdot 10^6$, the gas saturation curve exhibits a more gradual transition from the unsaturated to the saturated region. In contrast, for the difficult case of $P_r = 2 \cdot 10^3$ Pa, the gas saturation changes very quickly both at the injection point and at the interface with the saturated region. We note that the simulation results in figure 5.3 match well with those in [39, 62]. A comparison of the performance of the three solution methods is shown below in tables 5.2 and 5.3. TS, NS denote the total number of successful time steps and nonlinear iterations, respectively, and the numbers in parentheses are for the failed time steps and nonlinear iterations. Failed time steps are those in which the method diverges or does not converge

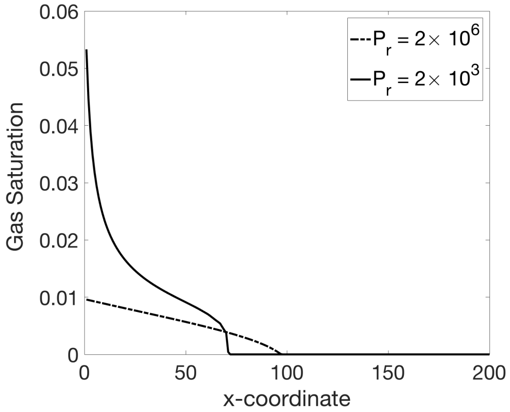


Figure 5.3: Gas infiltration into the domain for two different capillary pressure curves after 100,000 years.

within the allowed maximum number of iterations, and the failed nonlinear iterations correspond to those spent during the failed time steps. For both of these benchmark problems, an initial smoothing parameter $\tau = 10^{-6}$ and a reduction ratio $\beta = 0.1$ are used for the smooth Fischer-Burmeister approach.

For the MoMaS gas injection benchmark problem with $P_r = 2 \cdot 10^6$ Pa, the results in table 5.2 show that for the nonlinear solve, the Fischer-Burmeister function (without smoothing) does not show any improvement over the minimum function. It registers the same numbers of time steps needed to run the simulation both to 10^5 and to $5{\cdot}10^5$ years. In contrast, the smooth Fischer-Burmeister function achieves the same performance up to $T = 10^5$ years, and it reduces both the number of time steps and nonlinear iterations by about 20% for the full simulation. This suggests that the smooth Fischer-Burmeister function is better for simulating long time periods, when the gas phase infiltrates a larger portion of the domain.

The second example illustrates the effectiveness of the smooth Fischer-Burmeister

End Time (years)	min		FB		Smooth FB	
	TS	NS	TS	NS	TS	NS
10^5	5 (0)	35 (0)	5 (0)	35 (0)	5 (0)	36 (0)
$5 \cdot 10^5$	10 (0)	80 (0)	10 (0)	80 (0)	8 (0)	63 (0)

Table 5.2: Performance of the nonlinear solver for the capillary pressure model with $P_r = 2 \times 10^6$ Pa with mesh size of 200.

Mesh size	min		FB		Smooth FB	
	TS	NS	TS	NS	TS	NS
200	37 (20)	487 (195)	5 (0)	41 (0)	5 (0)	38 (0)
400	59 (48)	949 (440)	6 (0)	59 (0)	5 (0)	42 (0)

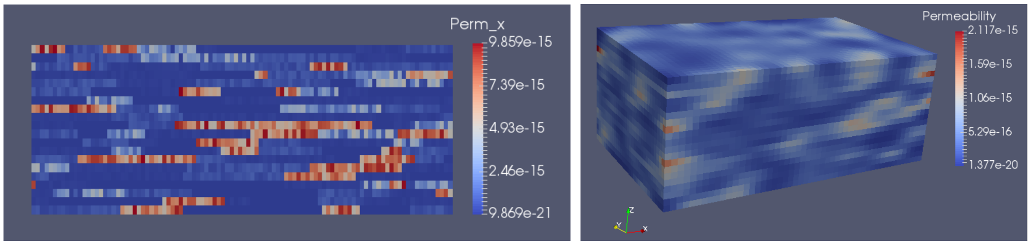
Table 5.3: Performance of the nonlinear solver for the highly nonlinear capillary pressure model with $P_r = 2 \times 10^3$ Pa after 100,000 years.

function in handling phase transitions for highly nonlinear problems. We compare the performance of the three different strategies and show the results in table 5.3. The semi-smooth Newton method with the minimum function struggles to converge for many time steps. It requires 37 and 59 time steps in total, with 20 and 48 failed time steps for mesh sizes of 200 and 400, respectively. Use of the Fischer-Burmeister function reduces the number of time steps by a factor of seven, and it also requires less than 10% number of nonlinear iterations. This means that on average, we can take about seven times larger time step and achieve approximately 90% decrease in run time with the Fischer-Burmeister function.

The approach using the smooth Fischer-Burmeister function registers about the same number of time steps as the approach using the standard Fischer-Burmeister function and it furthers decreases the number of time steps by 7% for the mesh size of 200. For the larger mesh of 400, however, the smooth Fischer-Burmeister variant shows a large improvement over the standard Fischer-Burmeister approach, requiring 29% fewer nonlinear iterations.

5.5.2 Problems with highly heterogeneous media

We perform two numerical experiments with highly heterogeneous permeability: (1) a modified two-dimensional SPE-10 problem, and (2) a three-dimensional problem. The permeability fields for these problems are shown in figures 5.4a and 5.4b. In both cases, the entry pressure for the Van Genuchten capillary pressure is chosen as $P_r = 2 \times 10^3$, which corresponds to the difficult nonlinear case in the pre-



(a) Modified two-dimensional SPE10 problem.

(b) Three-dimensional problem with random permeability.

Figure 5.4: Heterogeneous Problems.

vious benchmark problem. For the first case, we modify the two-dimensional SPE10 problem [28] by scaling the permeability field by a constant factor of 10^{-5} to make the porous medium more dense. The domain is a rectangle of size $762m \times 15.24m$. Pure hydrogen is injected on the left side $\Gamma_{\text{in}} = \{0\} \times [0, 15.24]$: $\psi^w \cdot \nu = 0$ and $\psi^h \cdot \nu = 5.57 \times 10^{-2} kg/m^2/year$, and a Dirichlet boundary condition is chosen on $\Gamma_{\text{out}} = \{762\} \times [0, 15.24]$: $P_l = 10^6$ Pa, $S_l = 1$, and $\rho_l^h = 0$. The upper and lower boundary is impervious, i.e. $\psi^w \cdot \nu = 0$ and $\psi^h \cdot \nu = 0$. Initial conditions are $P_l = 10^6$ Pa, $S_l = 1$, and $\rho_l^h = 0$ for the whole domain. For the spatial discretization, we use a 100×20 mesh. The initial time step $dt = 20$ days and the end time is $T_{\text{final}} = 1160$ days. The initial smoothing parameter for the smooth Fischer-Burmeister function is $\tau = 10^{-6}$.

In the second example, the domain is a box of size $50m \times 30m \times 20m$. The porosity and permeability fields are random, generated by a geostatistic model using the open-source code MRST [60]. The porosity has a range of $[0.002, 0.1]$ and the permeability varies from $1.377 \cdot 10^{-20}$ to $2.117 \cdot 10^{-15}$. Pure hydrogen is injected

	Two-dimensional SPE10		Three-dimensional problem	
C-function	FB	Smooth FB	FB	Smooth FB
Number of time steps	60 (7)	37 (4)	13 (4)	8 (3)
Average time step size (days)	19.3	31.3	151.8	250.0
Total nonlinear iterations	856 (147)	530 (84)	202 (84)	135 (63)
Execution time (s)	941.6	566.8	972.6	620.3

Table 5.4: Performance comparison for heterogeneous problems.

through the boundary at a corner: $\psi^w \cdot \nu = 0$ and $\psi^h \cdot \nu = 5.57 \times 10^{-2} kg/m^2/year$, and a Dirichlet boundary condition is chosen on the opposite corner: $P_l = 10^6$ Pa, $S_l = 1$, and $\rho_l^h = 0$. The rest of the boundary is impervious, i.e. $\psi^w \cdot \nu = 0$ and $\psi^h \cdot \nu = 0$. Initial conditions are $P_l = 10^6$ Pa, $S_l = 1$, and $\rho_l^h = 0$ for the whole domain. For the spatial discretization, we use a uniform $50 \times 30 \times 20$ mesh. The initial time step $dt = 200$ days and the end time is $T_{\text{final}} = 2000$ days. The initial smoothing parameter for the smooth Fischer-Burmeister function is $\tau = 10^{-4}$.

For both of these problems, the semi-smooth Newton approach using the minimum function fails to converge for many time steps, and dt becomes too small to obtain the full simulation results. Thus, only the results for the standard Fischer-Burmeister function and the smooth variant are reported in table 5.4. Again, the numbers in parentheses are for the failed time steps and nonlinear iterations. The Jacobian smoothing method combined with the smooth Fischer-Burmeister function is more robust than the semi-smooth Newton approach with the standard Fischer-Burmeister function, as demonstrated by the reduction in the number successful and failed time steps. For example, in the two-dimensional SPE10 problem, the former requires only 37 successful time steps and registers 4 failed time steps, as opposed to 60 successful and 7 failed time steps of the latter. In terms of performance, the Jacobian smoothing method combined with the smooth Fischer-Burmeister function is clearly better as it helps decrease the number of nonlinear iterations and execution time by 34-40% approximately for both the two-dimensional and three-dimensional problems.

5.5.3 Scaling Results

To study the parallel performance, we use the same setup as for the three-dimensional case with highly heterogeneous media considered in section 5.5.2. Parallel tests are run on Syrah, a Cray system with 5,184 Intel Xeon E5-2670 cores at the Lawrence Livermore National Laboratory Computing Center. Amanzi and other libraries are compiled with OpenMPI 1.6.5 and gcc-4.9.2. For strong scaling, the mesh size is fixed at $200 \times 120 \times 80$, and the problem has 5.76 million unknowns in total. We choose an initial time step of $dt = 2$ days and stop the simulation after 20 days. For weak scalability, the number of processors is increased in proportion to the problem size. We use meshes of size $50 \times 30 \times 20$, $100 \times 60 \times 40$, and $200 \times 120 \times 80$ with 2, 16, and 128 processors, respectively. The initial time step is set to $dt = 2$ days for all the mesh sizes and the simulation is stopped at $T = 200$ days. For both

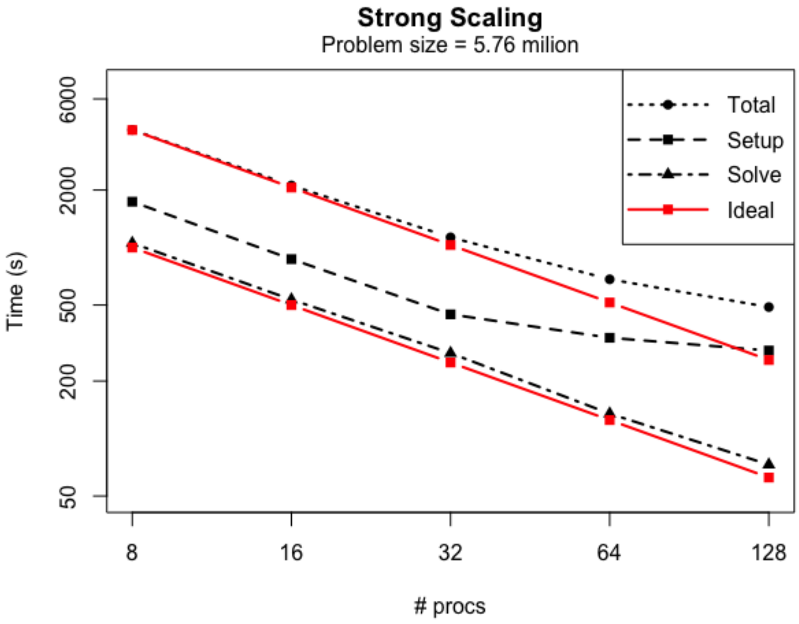


Figure 5.5: Strong scaling for the three-dimensional heterogeneous problem. The total runtime for the simulation, the setup, and solve time for the linear solver are reported.

cases, the entry pressure is set at $P_r = 2 \times 10^3$. The results reported in figure 5.5 show that the Jacobian smoothing method, combined with GMRES preconditioned by hypreMGR achieves near optimal strong scalability on 8 to 128 processors for the total time needed to run the whole simulation. The slight deviation from the ideal performance at 64 and 128 processors results from the decrease in parallel performance of the setup phase of hypreMGR, which has been observed in [22].

For weak scaling, a comparison between the Jacobian smoothing method using the smooth Fischer-Burmeister function and the semi-smooth Newton approach with the standard Fischer-Burmeister function is shown in tables 5.5 and 5.6. For the semi-smooth Newton method using the standard Fischer-Burmeister function,

Number of processors	2	16	128
Mesh size	$50 \times 30 \times 20$	$100 \times 60 \times 40$	$200 \times 120 \times 80$
Initial smoothing parameter τ	10^{-6}	10^{-6}	10^{-5}
Average step size (days)	28.6	28.6	25.0
Number of time steps	7	7	8
Average nonlinear iterations	5.1	6.6	8.9
Average linear iterations	10.7	13.5	17.4
Execution time	122 (s)	286 (s)	995 (s)

Table 5.5: Weak scaling performance of the Jacobian smoothing method.

Number of processors	2	16	128
Mesh size	$50 \times 30 \times 20$	$100 \times 60 \times 40$	$200 \times 120 \times 80$
Average step size (days)	28.6	25.0	3.45*
Number of time steps	7	8	11 (2)*
Average nonlinear iterations	4.7	6.6	11.1*
Average linear iterations	12.7	22.0	28.5*
Execution time	463 (s)	1623 (s)	> 4 hours

Table 5.6: Weak scaling performance of the semi-smooth Newton approach using the standard Fischer-Burmeister function.

the simulation does not finish within the 4-hour limit of run time on the cluster. Thus, we only report the solver statistics up to $T = 38$ days when the simulation terminates. As the mesh is refined, the Jacobian smoothing method is clearly more robust and efficient than the semi-smooth Newton method using the standard Fischer-Burmeister function. Not only does it reduce the number of nonlinear iterations, it also helps improve the performance of the linear solver as indicated by smaller number of linear iterations. The execution time is significantly reduced as a consequence.

5.6 Conclusions

In this work, we have developed a new Jacobian smoothing method based on the smooth Fischer-Burmeister function to solve the discrete nonlinear systems resulting from the the fully implicit discretization of the NCP formulation for compositional multiphase flow in porous media with phase transitions. Additionally, we performed various numerical experiments to compare our method with a semi-smooth Newton approach for two choices of C-function: the minimum and the Fischer-Burmeister functions. The results demonstrate that this method is significantly more robust and efficient with respect to the run time and number of nonlinear iterations. Unlike the semi-smooth Newton method using the minimum function, the Jacobian smoothing approach converges in all examples. Moreover, depending on the problem, it also reduces the number of nonlinear iterations and execution time by 34-40% compared to the semi-smooth Newton method using the standard

Fischer-Burmeister function.

Chapter 6: Concluding Remarks

With the rapid advance in computational power comes the increasing demand for high fidelity models that can take into account important effects, including capillarity, miscibility, and phase transitions. These effects result in the strong coupling between the physical variables such as pressure, saturation, and concentrations, and in most cases, fully coupled approaches are needed for fast convergence. However, fully coupled methods require solution of large, sparse linear systems that are particularly difficult to solve when capillarity and phase transitions are present. For flow in highly heterogeneous media, phase transitions also causes slow convergence for even state-of-the-art nonlinear solvers, such as semi-smooth Newton's methods for NCP reformulation of compositional multiphase flow. This dissertation aims to address these computational issues through the design of robust and scalable AMG preconditioners for iterative methods applied to the linear systems arising from discretizations of the fully coupled incompressible and compositional two-phase flow models, as well as development of a new method for the nonlinear solve for compositional two-phase flow model with phase transitions.

In particular, in chapter 3, we implemented a parallel simulator for the fully coupled incompressible two-phase flow model and developed a new block factorization preconditioner that can capture the effect of capillary pressure on the flow. An analysis of the eigenvalues of the preconditioned system using the block factorization method showed that all the eigenvalues are clustered and bounded away from zero. The effectiveness and scalability of the preconditioner were demonstrated for both advection-dominated and diffusion-dominated flows on a wide range of test problems, including those with highly heterogeneous media such as the three-dimensional SPE10 problem.

We turned our attention to compositional models with phase transitions in chapter 4. We implemented a simplified two-phase, two-component flow model using the NCP formulation, because this method shows excellent robustness in handling phase transitions, and by avoiding variable switching, it uses a fixed set of primary variables. We then develop hypreMGR, an AMG preconditioner based on multigrid reduction for GMES applied to the indefinite linear systems that result from linearization of the NCP formulation using a semi-smooth Newton's method. By choosing an appropriate coarse grid for each level of reduction, our method can handle the presence of zeros on the diagonal of the linear systems when phase transitions are present. We demonstrated that our method is applicable to various models of capillary pressure, and it also scales optimally with problem size of up to 100 million unknowns and 1024 processors.

Finally in chapter 5, we addressed the robustness issue of the semi-smooth Newton's methods for problems with phase transitions in highly heterogeneous media. We developed a new Jacobian smoothing method using a smooth approximation of the Fischer-Burmeister function to solve the NCP formulation of the simplified two-phase, two-component flow with phase transitions. Our numerical results showed that the nonlinear Fischer-Burmeister function is a better choice than the piecewise linear minimum function as a C-function for the NCP formulation that is solved with a semi-smooth Newton's method. We demonstrated that the Jacobian smoothing method using the smooth version of the Fischer-Burmeister is more efficient than both of the semi-smooth Newton's methods that use the minimum and the Fischer-Burmeister functions, especially for problems in highly heterogeneous media.

Bibliography

[1] A. Abadpour and M. Panfilov. Method of negative saturations for modeling two-phase compositional flow with oversaturated zones. *Transport in Porous Media*, 79(2):197–214, 2008.

[2] G. Acs, S. Doleschall, and E. Farkas. General purpose compositional model. *Society of Petroleum Engineers Journal*, 25(04):543–553, 1985.

[3] M. Aganagić. Newton's method for linear complementarity problems. *Mathematical Programming*, 28(3):349–362, 1984.

[4] M. B. Allen. Numerical modelling of multiphase flow in porous media. *Advances in Water Resources*, 8(4):162–187, 1985.

[5] ASCEM. `http://esd.lbl.gov/research/projects/ascem/thrusts/hpc/`, 2009.

[6] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, 1994.

[7] K. Aziz and A. Settari. *Petroleum Reservoir Simulation.* Applied Science Publishers, 1979.

[8] R. E. Bank and R. K. Smith. The incomplete factorization multigraph algorithm. *SIAM Journal on Scientific Computing*, 20(4):1349–1364, 1999.

[9] P. Bastian. Numerical computation of multiphase flow in porous media. Habilitationsschrift, 1999.

[10] P. Bastian and R. Helmig. Efficient fully-coupled solution techniques for two-phase flow in porous media: Parallel multigrid solution and large scale computations. *Advances in Water Resources*, 23(3):199 – 216, 1999.

[11] J. Bear. *Dynamics of Fluids in Porous Media.* American Elsevier Publishing Company, Inc., New York, 1972.

[12] A. Behie and P. K. W. Vinsome. Block iterative methods for fully implicit reservoir simulation. *Society of Petroleum Engineers Journal*, 22(05):658–668, 1982.

[13] I. Ben Gharbia and J. Ch. Gilbert. Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134:349–364, 2012.

[14] I. Ben Gharbia and J. Ch. Gilbert. An algorithmic characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3):904–916, 2013.

[15] I. Ben Gharbia and J.Ch. Gilbert. An algorithmic characterization of P-matricity II: corrections, refinements, and validation. *SIAM Journal on Matrix Analysis and Applications* (submitted), 2018. [preprint].

[16] I. Ben Gharbia and J. Jaffré. Gas phase appearance and disappearance as a problem with complementarity constraints. *Mathematics and Computers in Simulation*, 99(0):28 – 36, 2014.

[17] M. Benzi and A. J. Wathen. Some preconditioning techniques for saddle point problems. In *Mathematics in Industry*, volume vol 13, pages 195–211. Springer, Berlin, Heidelberg, 2008.

[18] A. Bourgeat, M. Jurak, and F. Smaï. Two phase partially miscible ow and transport modeling in porous media; application to gas migration in a nuclear waste repository. *Computational Geosciences*, 13(1):29–42, 2008.

[19] A. Bourgeat, F. Smaï, and S. Granet. Compositional Two-Phase Flow in Saturated-Unsaturated Porous Media: Benchmarks for Phase Appearance/Disappearance. In *Simulation of Flow in Porous Media*, pages 81–106. De Gruyter, 2013.

[20] R. Brooks and A. Corey. *Hydraulic Properties of Porous Media.* Hydrology Papers, Colorado State Univeristy, 1964.

[21] 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):S662S680, 2017.

[22] 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.

[23] G. Chavent and J. Jaffré. *Mathematical Models and Finite Elements for Reservoir Simulation: Single Phase, Multiphase and Multicomponent Flows Through Porous Media (Studies in Mathematics & its Applications)*. Elsevier Science Ltd, 1986.

[24] B. Chen and P. T. Harker. A non-interior-point continuation method for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1168–1190, 1993.

[25] Z. Chen and R. E. Ewing. Comparison of various formulations of three-phase flow in porous media. *Journal of Computational Physics*, 132(2):362 – 373, 1997.

[26] Z. Chen, G. Huan, and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media*. Society for Industrial and Applied Mathematics, 2006.

[27] E. Chow and P. S. Vassilevski. Multilevel block factorizations in generalized hierarchical bases. *Numerical Linear Algebra with Applications*, 10(1-2):105–127, 2003.

[28] 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 (SPE), 2001.

[29] T. J. Chung. *Applied Continuum Mechanics*. Cambridge University Press, 1996.

[30] H. Class, A. Ebigbo, R. Helmig, H. K. Dahle, J. M. Nordbotten, M. A. Celia, P. Audigane, M. Darcis, J. Ennis-King, Y. Fan, B. Flemisch, S. E. Gasda, M. Jin, S. Krug, D. Labregere, A. N. Beni, R. J. Pawar, A. Sbai, S. G. Thomas, L. Trenty, and L. Wei. A benchmark study on problems related to CO2 storage in geologic formations. *Computational Geosciences*, 13(4):409–434, 2009.

[31] H. Class and R. Helmig. Numerical simulation of non-isothermal multiphase multicomponent processes in porous media. 2. Applications for the injection of steam and air. *Soil Science Society of America Journal*, 44:892–898, 2002.

[32] A. J. Cleary, R. D. Falgout, V. E. Henson, and J. E. Jones. Coarse-grid selection for parallel algebraic multigrid. In Alfonso Ferreira, Jos Rolim, Horst Simon, and Shang-Hua Teng, editors, *Solving Irregularly Structured Problems in Parallel*, volume 1457 of *Lecture Notes in Computer Science*, pages 104–115. Springer Berlin Heidelberg, 1998.

[33] T. Clees and L. Ganzer. An efficient algebraic multigrid solver strategy for adaptive implicit methods in oil-reservoir simulation. *SPE Journal*, 15(03):670–681, 2010.

[34] K. H. Coats. An equation of state compositional model. *SPE Journal*, 20(05):363–376, 1980.

[35] E. Coon, J. D. Moulton, and S. Painter. Managing complexity in simulations of land surface and near-surface processes. Technical Report LA-UR 14-25386, Applied Mathematics and Plasma Physics Group, Los Alamos National Laboratory, 2014.

[36] A. T. Corey. *Mechanics of Immiscible Fluids in Porous Media.* Water Resources Publications, 1995.

[37] M. Cusini, A. A. Lukyanov, J. Natvig, and H. Hajibeygi. Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. *Journal of Computational Physics*, 299(Supplement C):472 – 486, 2015.

[38] C. N. Dawson, H. Klie, M. F. Wheeler, and C. S. Woodward. A parallel, implicit, cell-centered method for two-phase flow with a preconditioned Newton-Krylov solver. *Computational Geosciences*, 1(3-4):215–249, 1997.

[39] R. De Cuveland. *Two-Phase Compositional Flow Simulation with Persistent Variables.* PhD thesis, University of Heidelberg, 2015.

[40] J. Douglas and D. W. Peaceman. Numerical solution of two-dimensional heat-flow problems. *AIChE Journal*, 1(4):505–512, 1955.

[41] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–421, 1956.

[42] H. C. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27(5):1651–1668, 2006.

[43] H. C. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 227(3):1790–1808, 2008.

[44] B. Engquist, P. Lötstedt, and O. Runborg, editors. *Multiscale Modeling and Simulation in Science: 66 (Lecture Notes in Computational Science and Engineering)*. Springer, 2009.

[45] B. Engquist and B. Sjögreen. The convergence rate of finite difference schemes in the presence of shocks. *SIAM Journal on Numerical Analysis*, 35(6):2464–2485, dec 1998.

[46] F. Facchinei and J. S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems (Springer Series in Operations Research and Financial Engineering)*, volume 2. Springer, 2003.

[47] R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing*, 36(6):C635–C661, 2014.

[48] R. D. Falgout, J. E. Jones, and U. M. Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. In AreMagnus

Bruaset and Aslak Tveito, editors, *Numerical Solution of Partial Differential Equations on Parallel Computers*, volume 51 of *Lecture Notes in Computational Science and Engineering*, pages 267–294. Springer Berlin Heidelberg, 2006.

[49] R.D. Falgout and U.M. Yang. HYPRE: a library of high performance preconditioners. In *Preconditioners, Lecture Notes in Computer Science*, pages 632–641, 2002.

[50] M. Ferronato, C. Janna, and G. Pini. A generalized block FSAI preconditioner for nonsymmetric linear systems. *Journal of Computational and Applied Mathematics*, 256(Supplement C):230 – 241, 2014.

[51] P. A. Forsyth, Y-S. Wu, and K. Pruess. Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media. *Advances in Water Resources*, 18(1):25–38, 1995.

[52] V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2000.

[53] M. A. Heroux, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, K. S. Stanley, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, and R. P. Pawlowski. An overview of the Trilinos project. *ACM Transactions on Mathematical Software*, 31(3):397–423, 2005.

[54] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[55] U. Hornung. *Homogenization and Porous Media (Interdisciplinary Applied Mathematics)*. Springer, 2012.

[56] C. Kanzow. Some noninterior continuation methods for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 17(4):851–868, 1996.

[57] H. Klie, M. Rame, and M. F. Wheeler. Two-stage preconditioners for inexact Newton methods in multiphase reservoir simulation. Technical report, Center for Research on Parallel Computation, Rice University, 1996.

[58] S. Lacroix, Y. 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.

[59] A. Lauser, C. Hager, R. Helmig, and B. Wohlmuth. A new approach for phase transitions in miscible multi-phase flow in porous media. *Advances in Water Resources*, 38:957–966, 2011.

[60] K. A. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. M. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322, 2011.

[61] B. Lu. *Iteratively Coupled Reservoir Simulation for Multiphase Flow in Porous Media*. PhD thesis, Center for Subsurface Modeling, ICES, The University of Texas - Austin, 2008.

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

[63] E. Marchand, T. Müller, and P. Knabner. Fully coupled generalised hybrid-mixed finite element approximation of two-phase two-component flow in porous media. Part II: numerical scheme and numerical results. *Computational Geosciences*, 16(3):691–708, 2012.

[64] M. L. Michelsen. The isothermal flash problem. Part II. Phase-split calculation. *Fluid Phase Equilibria*, 9(1):21 – 40, 1982.

[65] R. Neumann, P. Bastian, and O. Ippisch. Modeling and simulation of two-phase two-component flow with disappearing nonwetting phase. *Computational Geosciences*, 17(1):139–149, 2013.

[66] J. Nocedal and S. Wright. *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, 2006.

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

[68] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787 – 1806, 1972.

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

[70] D. W. Peaceman and H. H. Rachford Jr. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):28–41, 1955.

[71] L. Qi. Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18:227–244, 1993.

[72] L. Qi and D. Sun. *Progress in Optimization: Contributions from Australasia*, chapter A Survey of Some Nonsmooth Equations and Smoothing Newton Methods, pages 121–146. Springer US, Boston, MA, 1999.

[73] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–367, 1993.

[74] C. Qiao, S. Wu, J. Xu, and C. S. Zhang. Analytical decoupling techniques for fully implicit reservoir simulation. *Journal of Computational Physics*, 336(Supplement C):664 – 681, 2017.

[75] A. Reusken. A multigrid method based on incomplete Gaussian elimination. *Numerical Linear Algebra with Applications*, 3(5):369–390, 1996.

[76] A. Reusken. On the approximate cyclic reduction preconditioner. *SIAM Journal on Scientific Computing*, 21(2):565–590, 1999.

[77] L. A. Richards. Capillary conduction of liquids through porous mediums. *Physics*, 1(5):318–333, 1931.

[78] M. Ries and U. Trottenberg. MGR-Ein blitzschneller elliptischer Löser. *Preprint 277SBF 72, Universität Bonn*, 1979.

[79] M. Ries, U. Trottenberg, and G. Winter. A note on MGR methods. *Linear Algebra and its Applications*, 49:1 – 26, 1983.

[80] J. W. Ruge and K. Stüben. 4. Algebraic Multigrid. In *Multigrid Methods*, pages 73–130. Society for Industrial and Applied Mathematics, 1987.

[81] T. F. Russell and M. F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. In *The Mathematics of Reservoir Simulation*, chapter 2, pages 35–106. Society for Industrial and Applied Mathematics, 1983.

[82] T. Rusten and R. Winther. A preconditioned iterative method for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 13(3):887–904, 1992.

[83] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, second edition, 2003.

[84] 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.

[85] S. W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *International Journal for Numerical Methods in Engineering*, 23(2):239–251, 1986.

[86] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(12):281 – 309, 2001.

[87] 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 (SPE), 2007.

[88] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, Inc., Orlando, FL, USA, 2001.

[89] J. D. Valiantzas. Combined Brooks-Corey/Burdine and van Genuchten/Mualem closed-form model for improving prediction of unsaturated conductivity. *Journal of Irrigation and Drainage Engineering*, 137(4):223–233, 2011.

[90] 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.

[91] M. van Genuchten. A closed form equation for predicting the hydraulic conductivity of unsaturated soils. 25:551–564, 1980.

[92] C. Wagner, W. Kinzelbach, and G. Wittum. Schur-complement multigrid a robust method for groundwater flow and transport problems. *Numerische Mathematik*, 75:523 – 545, 1997.

[93] J. R. Wallis, R. P. Kendall, and L. E. Little. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers (SPE), 1985.

[94] L. Wang, D. Osei-Kuffuor, R. D. Falgout, I. D. Mishev, and J. Li. Multigrid reduction for coupled flow problems with application to reservoir simulation. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers (SPE), SPE-182723-MS, 2017.

[95] J. W. Watts. A compositional formulation of the pressure and saturation equations. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers (SPE), SPE 12244, 1985.

[96] S. Whitaker. Flow in porous media I: A theoretical derivation of Darcy's law. *Transport in Porous Media*, 1(1):3–25, 1986.

[97] J. A. White and R. I. Borja. Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics. *Computational Geosciences*, 15(4):647–659, 2011.

[98] Y-S. Wu and P. A. Forsyth. On the selection of primary variables in numerical formulation for modeling multiphase flow in porous media. *Journal of Contaminant Hydrology*, 48:277–304, 2001.

[99] H. Yang, S. Sun, and C. Yang. Nonlinearly preconditioned semismooth Newton methods for variational inequality solution of two-phase flow in porous media. *Journal of Computational Physics*, 332(Supplement C):1 – 20, 2017.

[100] U. M. Yang. Parallel algebraic multigrid methods - high-performance preconditioners. In *Numerical Solution of Partial Differential Equations on Parallel Computers*, volume 51 of *Lecture Notes in Computational Science and Engineering*, pages 209–236. Springer Berlin Heidelberg, 2006.

[101] L. C. Young and R. E. Stephenson. A generalized compositional approach for reservoir simulation. *SPE Journal*, 23(05):727–742, 1983.