SCALABLE LINEAR AND NONLINEAR ALGORITHMS FOR MULTIPHASE FLOW IN POROUS MEDIA

A DISSERTATION
SUBMITTED TO THE PROGRAM IN SCIENTIFIC COMPUTING
AND COMPUTATIONAL MATHEMATICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Wing Hong Felix Kwok

December 2007

© Copyright by Wing Hong Felix Kwok 2008
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Hamdi Tchelepi) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Khalid Aziz)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Michael Saunders)

Approved for the University Committee on Graduate Studies.

Abstract

The efficient simulation of immiscible fluid displacements in underground porous media remains an important and challenging problem in reservoir engineering. First, the governing PDEs exhibit a mixed hyperbolic-parabolic character due to the coupling between the global flow and the local transport of the different phases. The transport problem is highly nonlinear, leading to the formation of shock fronts and steep gradients in the saturation profile. In addition, rock properties such as porosity and permeability are highly heterogeneous, leading to poor numerical conditioning of the resulting linear systems. Finally, fluid velocities vary greatly across the domain, with near-well regions experiencing fast flows and some far away regions experiencing almost no flow at all. Consequently, the use of explicit integrators would entail a time-step restriction that is much more severe than the global reservoir time scales. For this reason, implicit time-stepping is the preferred temporal discretization in the reservoir simulation community, but this requires the solution of a very large system of nonlinear algebraic equations (often on the order of millions of unknowns) at each time step.

Our main algorithmic contribution is the ordering of equations and unknowns in such a way that flow directions are exploited. This leads to improvements in both the linear and nonlinear solvers. In the nonlinear setting, the ordering leads to a reduced-order Newton method, which numerical experiments have shown to have a much more robust convergence behavior than the usual Newton's method. We also prove, for 1D incompressible two-phase flow, that the reduced Newton method converges for any time-step size. In the linear solver, ordering improves the convergence of the Constrained Pressure Residual (CPR) preconditioner and reduces its sensitivity to flow configurations.

We also present a rigorous analysis of phase-based upstream discretization, which is different from the classical Godunov and Engquist-Osher schemes for nonlinear conservation laws. We show, based on a fully nonlinear analysis, that the fully implicit scheme is well-defined, stable, monotonic and converges to the entropy solution for arbitrary CFL numbers. Thus, unlike the existing linear stability analysis, our results provide a rigorous justification for the empirical observation that fully-implicit solutions are always stable and yield monotonic profiles.

Acknowledgement

I would like to express my utmost gratitude towards my advisor, Prof. Hamdi Tchelepi, not only for his insights and guidance, but also for his patience and encouragement when things did not go so well. This research would not have been possible without his constant input and moral support. In addition, I would like to thank Prof. Khalid Aziz for making many useful suggestions. I am also indebted to Prof. Margot Gerritsen, who introduced me to porous media flow, and to Philipp Birken, who gave constructive comments on several chapters of this dissertation.

Much appreciation goes to my office mates, Rami Younis and Marc Hesse, for our fruitful and entertaining conversations. Our interactions were thoroughly enriching both on a professional and a personal level, and I will really miss your company. I also thank Yuanlin Jiang and Huanquan Pan for their help with GPRS-related issues.

Many thanks to Prof. Michael Saunders, who agreed to join the reading committee at the very last minute and did an amazingly thorough job as a reader. Thanks also to Indira Choudhury, who helped me tremendously in putting my orals commitee back together when it almost fell apart. Finally, I am grateful for the moral support from my parents, who always believed in me throughout this rather long journey.

This dissertation is dedicated to the memory of Prof. Gene Golub, who passed away two weeks before my PhD defense. I will always remember him for the depth of his knowledge on all aspects of scientific computing, as well as his generosity and genuine interest in the well-being of every student in SCCM/ICME. He is truly an irreplaceable figure in our community, and he will be sorely missed.

I would like to thank the SUPRI-B reservoir simulation affliates program for its financial support for this research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Petroleum reservoir simulation is the use of numerical techniques to solve the equations for heat and fluid flow in porous media, given the appropriate initial and boundary conditions. Simulation technology has evolved tremendously since the development of the first simulator in the 1950s. Due to the explosion of available computing power and the ever-increasing sophistication of simulation techniques, simulation has become an indispensible tool to reservoir engineering. Today, nearly all major reservoir development decisions are based at least partially on simulation results [83]. Despite the growing speed and storage capacities of today's computers, there is increasing interest and necessity to simulate larger and more complex reservoir models. As a result, the efficient simulation of miscible and immiscible fluid displacements in underground porous media remains an important and challenging problem in reservoir engineering.

There are several hurdles to an efficient, scalable reservoir simulator. First, the governing PDEs exhibit a mixed hyperbolic-parabolic character due to the coupling between the flow (pressure and total velocity) and the transport (phase saturations) problems. In addition, rock properties such as porosity and permeability are highly heterogeneous, leading to poor numerical conditioning of the resulting linear systems. Finally, fluid velocities vary greatly across the domain, with near-well regions experiencing fast flows and some far away regions experiencing almost no flow at all. These characteristics impose severe constraints on the numerical methods used in practical reservoir simulation. In particular, scalable techniques that work well for specific classes of problems (e.g., algebraic multigrid for elliptic problems [74]) no longer work well for reservoir simulation problems.

The simplest and most widely used model in reservoir simulation is the standard black-oil model [6]. In this model, mass transfer between the hydrocarbon liquid and vapor phases is represented using pressure-dependent solubilities, and the compressibility effects are represented using normalized densities (the so-called formation volume factors). These simplifying assumptions on fluid properties are used to eliminate the need for equation of state (EOS) and phase equilibrium calculations, which can take up to 70% of the total simulation time [87, 16]. Thus, despite the increasing use of compositional models, black-oil simulation still accounts for the vast majority of simulations in industry. Hence, this thesis will concentrate on improving the efficiency and robustness of black oil simulation.

The rest of this chapter is organized as follows. In section 1.1, we derive the PDEs that describe the black-oil model. In section 1.2, we introduce the finite-volume discretization, as well as the various time-marching schemes that are used to integrate the PDEs in time. We also describe the most commonly used methods to solve the resulting system of nonlinear and linear equations. We outline the remainder of the thesis and state our contributions in section 1.3.

1.1 Governing equations

1.1.1 General mass-balance equations

The governing equations for multiphase flow in porous media are based on the conservation of mass for each *component*. Here, a component can be either a single chemical species (e.g., decane $C_{10}H_{22}$), or a mixture of components that behave similarly, so that they can be lumped together into a pseudocomponent. When n_c components are present, the system of conservation laws has the form

$$\frac{\partial c_i}{\partial t} + \nabla \cdot \mathbf{F}_i = q_i, \quad i = 1, \ldots, n_c, \tag{1.1.1}$$

where c_i is the mass concentration of component i, F_i is the mass flux and q_i is the source or sink term. Each component can exist in one or more immiscible fluid *phases* that flow inside the pore space; typically, we consider either two-phase (aqueous and liquid hydrocarbon) or three-phase (aqueous, liquid and vapor hydrocarbon) flow problems. If X_{ij} is the concentration of component i in phase j (mass per unit volume), then the concentration of component i can be written as

$$c_i = \phi \sum_{j=1}^{n_p} X_{ij} S_j, \tag{1.1.2}$$

where $\phi = \phi(x)$ is the porosity of the medium (i.e., the fraction of the bulk volume that is open to fluid flow), n_p is the number of phases present, and S_j is the saturation of phase j (i.e., the fraction of the pore volume occupied by phase j). The mass flux $\mathbf{F}_i$ is the sum of the volumetric fluxes of each phase j, multiplied by the concentration X_{ij}. In other words,

$$\mathbf{F}_i = \sum_{j=1}^{n_p} X_{ij} \mathbf{u}_j, \tag{1.1.3}$$

where $\mathbf{u}_j$ is the volumetric flux vector of phase j. The volumetric fluxes are given by *generalized Darcy's law*:

$$\mathbf{u}_j = -\frac{k_{rj}}{\mu_j} \mathbf{K}(\nabla p_j - \gamma_j \nabla z), \tag{1.1.4}$$

where $\mathbf{K}$ is the absolute permeability tensor, z is the depth variable; and for each phase j, $k_{rj} = k_{rj}(S_1, \ldots, S_{n_p})$ is the relative permeability of phase j, μ_j is the phase viscosity, p_j is the phase pressure, and γ_j is the gravitational force acting on phase j. The permeability tensor $\mathbf{K}$ is highly variable over the domain, even within short distances; it also exhibits complex correlation patterns over a hierarchy of spatial scales. For simulation purposes, it is generally necessary to assume $\mathbf{K}$ to be a discontinuous function of $\mathbf{x}$, since it would be impractical (or even impossible) to simulate on a scale over which $\mathbf{K}$ becomes continuous. This has implications on the choice of spatial discretization, which is described in section 1.2.1.

We also have a few algebraic constraints in addition to the above PDEs. Since the pore space is saturated, we have the constraint

$$\sum_{j=1}^{n_p} S_j = 1, \tag{1.1.5}$$

and the phase pressures are related by the *capillary pressure constraints*:

$$p_j - p_{j+1} = P_{cj,j+1}(S_1, \dots, S_{n_p}), \quad j = 1, \dots, n_p - 1. \tag{1.1.6}$$

1.1.2 Black-oil model

Equations (1.1.1), (1.1.5) and (1.1.6) yield $n_c + n_p$ equations, and we have $2n_p$ unknowns corresponding to the phase pressures and saturations. In a compositional model, the concentrations X_{ij} are also treated as unknowns, and additional equations are needed to close the system (cf. [58]). However, for the black-oil model, we have $n_c = n_p$, and X_{ij} are treated as known functions of p_j, so that we have the same number of equations and unknowns. Specifically, the black-oil assumptions are as follows:

1. The chemical species are represented by three pseudocomponents: water, oil and gas, which are aligned with the aqueous, liquid and vapor hydrocarbon phases respectively;

2. The water component exists only in the aqueous phase, and the oil component exists only in the liquid hydrocarbon phase;

3. The gas component can exist in both the liquid and vapor hydrocarbon phases, but gas solubility in the liquid phase is a pure function of p_g (the vapor-phase pressure).

With these assumptions, the mass-balance equations (1.1.1) take the form

$$\frac{\partial(\phi \rho_p S_p)}{\partial t} + \nabla \cdot (\rho_p \mathbf{u}_p) = \rho_p q_p \tag{1.1.7}$$

for $p = o, w$ (liquid and aqueous phases), and

$$\left(\frac{\partial(\rho_g \phi S_g)}{\partial t} + \nabla \cdot (\rho_g \mathbf{u}_g)\right) + \left(\frac{\partial(\rho_o \phi S_o R_s)}{\partial t} + \nabla \cdot (\rho_o \mathbf{u}_o R_s)\right) = \rho_g q_g \tag{1.1.8}$$

for the vapor phase, where $R_s = R_s(p_g)$ is the solubility ratio. The generalized Darcy's law (1.1.4), which is valid for $p = w, o, g$, is used to obtain the phase velocities, $\mathbf{u}_p$. In practical simulations, we typically rewrite the PDEs in terms of a set of linearly independent *primary variables* (usually S_w, S_g and p_g, but one can choose any phase pressure and any $n_p - 1$ saturations), and then use the algebraic relations (1.1.5) and (1.1.6) to calculate the remaining variables. In addition, it is commonly assumed that the relative permeabilities k_{rp} and capillary pressures P_{cpq} have the following dependencies on saturation:

$$k_{rw} = k_{rw}(S_w), \qquad k_{ro} = k_{ro}(S_w, S_g), \qquad k_{rg} = k_{rg}(S_g); \tag{1.1.9}$$

$$p_o - p_w = P_{cow}(S_w), \qquad p_g - p_o = P_{cgo}(S_g). \tag{1.1.10}$$

The above functions are all nonlinear with respect to the saturation variables, and they contribute to the highly nonlinear character of the resulting PDEs [71]. The parameterization is based on the assumption that water is the most wetting phase and gas the least wetting phase, which is valid for most reservoirs of interest (see [6] for more detailed explanations). We also need $P'_{cow} \leq 0$ and $P'_{cgo} \geq 0$ for the problem to be well-posed. The resulting system of PDEs is supplemented with the boundary conditions

$$p_w = p_{wd} \quad \text{on } \Gamma_d \tag{1.1.11}$$

$$\rho_w \mathbf{u}_w \cdot \boldsymbol{\nu} = g_{wn} \quad \text{on } \Gamma_n \tag{1.1.12}$$

$$\rho_o \mathbf{u}_o \cdot \boldsymbol{\nu} = g_{on} \quad \text{on } \Gamma_n \tag{1.1.13}$$

$$\rho_g \mathbf{u}_g \cdot \boldsymbol{\nu} = g_{gn} \quad \text{on } \Gamma_n \tag{1.1.14}$$

and initial conditions

$$p_w(x,0) = p_{w0}(x), \quad S_w(x,0) = S_{w0}(x), \quad S_g(x,0) = S_{g0}(x), \tag{1.1.15}$$

where the Dirichlet boundary Γ_d has positive measure, and ν denotes the outward normal to the boundary.

Incompressible flow (and other simplifications)

In subsequent chapters, we often consider the case of incompressible flow, which implies the phases have constant densities ρ_p. For simplicity, we also restrict our attention to heterogeneous, but pointwise isotropic permeabilities, i.e., $\mathbf{K} = K\mathbf{I}$, where $\mathbf{I}$ is the identity tensor. In this case, the conservation equations become

$$\phi \frac{\partial S_p}{\partial t} - \nabla \cdot (\lambda_p K \nabla (p_p - \gamma_p z)) = q_p \tag{1.1.16}$$

for $p = o, w$, and

$$\left(\phi \frac{\partial S_g}{\partial t} - \nabla \cdot \left[\lambda_g K \nabla (p_g - \gamma_g z) \right] \right) + \left(\phi \frac{\partial (S_o \bar{R}_s)}{\partial t} - \nabla \cdot \left[\bar{R}_s \lambda_o K \nabla (p_o - \gamma_o z) \right] \right) = q_g, \tag{1.1.17}$$

for the gas phase, where $\lambda_p = k_{rp}/\mu_p$ is the (relative) mobility of phase p, and $\bar{R}_s = \rho_o R_s / \rho_g$ is the normalized solubility ratio. Sometimes we also consider the two-phase flow case, which is simply the same PDEs with the gas-related equations removed.

Pressure equation

An important equation that can be derived from the mass balance equations and the saturation constraint is the *pressure equation*. It can be obtained by taking a special linear combination of the mass-balance equations (1.1.7), (1.1.8). Assume there are no source or sink terms and no buoyancy effects, and suppose $P_{cow} = P_{cgo} = 0$, so that all the phase pressures are identical. Inclusion of such terms would introduce additional lower order terms, but would not alter the fundamental character of the PDE. Let us multiply the water equation by $1/\rho_w$, the gas equation by $1/\rho_g$, and the oil equation by $(1 - \bar{R}_s)/\rho_o$. Assuming that the pressure p is differentiable and that ϕ, ρ_p and R_s are smooth functions of pressure, we get (after some algebra):

$$\phi c_T \frac{\partial p}{\partial t} - \nabla \cdot (\lambda_T K \nabla p) - K \chi_T |\nabla p|^2 = 0, \tag{1.1.18}$$

where the phase compressibilities are

$$\begin{aligned} c_w &= \frac{\rho_w'}{\rho_w}, \qquad & c_o &= \frac{\rho_o'}{\rho_o} + \frac{\rho_o R_s'}{\rho_g}, \\ c_g &= \frac{\rho_g'}{\rho_g}, \qquad & c_r &= \frac{\phi'}{\phi}, \end{aligned}$$

and the 'total' quantities (denoted with the subscript T) are

Total compressibility:	$c_T = S_w c_w + S_o c_o + S_g c_g + c_r,$
Total mobility:	$\lambda_T = \lambda_w + \lambda_o + \lambda_g,$
Mobility-weighted compressibility:	$\chi_T = \lambda_w c_w + \lambda_o c_o + \lambda_g c_g.$

The full derivation is shown in Appendix A. Equation (1.1.18) is a parabolic PDE in p with an additional quadratic nonlinear term $K\chi_T|\nabla p|^2$. We must have $c_T > 0$ for the problem to be well posed. (This criterion has been exploited by Coats in [20] to derive validity checks for PVT data of isothermal black-oil and compositional fluid systems.) An analytic solution can be found for the constant-coefficient analog of (1.1.18):

$$\begin{gathered} \frac{\partial u}{\partial t} - a\nabla^2 u + b|\nabla u|^2 = 0, \quad (x,t) \in \mathbb{R}^n \times (0, \infty) \\ u(x,0) = g, \end{gathered}$$

where $a > 0$ and b are constants [32, §4.4]. When $c_T \equiv 0$ (the incompressible case), (1.1.18) degenerates to an elliptic equation in p:

$$-\nabla \cdot (K \lambda_T \nabla p) = 0. \tag{1.1.19}$$

The pressure equation is important because it dictates the choice of numerical methods and forms the basis for several widely used methods in reservoir simulation.

1.2 Numerical simulation of the reservoir

In order to simulate fluid flow in the reservoir, the above governing equations need to be discretized in time and space, and the resulting systems of nonlinear algebraic equations need to be solved at every time step. A *reservoir simulator*, which integrates the governing equations up to a final time T_{final} based on given initial conditions, will typically follow these steps during the simulation process:

1. Read input data (model grid geometry, permeability, porosity, fluid properties, etc.);
2. Initialize reservoir (initial conditions, equilibrium calculations);
3. Set boundary conditions;
4. While T_{final} not reached:
   - Compute an appropriate Δt;
   - Set well locations and production/injection rates for the current time step;
   - Form the nonlinear algebraic equations that arise from discretizing the governing equations;
   - Solve the nonlinear system;
   - Print results (water cut, saturation profile, etc.) if necessary;
   - Increment time;
5. End when T_{final} is reached.

A robust general-purpose simulator needs to handle a variety of reservoir specifications (model sizes, property distributions) and flow configurations. It is this generality requirement that dictates the choice of numerical methods that are used to approximate the PDEs. In this section, we provide the background for the remainder of the thesis by briefly discussing several common discretizations and solvers; for a broader survey of discretizations that are used in reservoir simulation, we direct the reader to [34, 52, 83]. A discussion of time-step control and the treatment of wells is beyond the scope of this thesis, even though these are very important considerations in building an accurate and useful simulator (see [6] for details).

1.2.1 Spatial discretization

Historically, the majority of reservoir simulators used (and still use) finite volume methods to discretize the multiphase flow equations. This choice is motivated by the need for exact local conservation, since shocks will generally be present in the saturation profile in the incompressible case. When compressibility and capillarity are present, the analytical solution will no longer contain shocks, but steep gradients will remain in the saturation profile, and it would be computationally costly to use a grid that is fine enough to resolve these gradients. The discretized component mass-balance equations are written in conservation form:

$$\frac{\partial(\phi_i \rho_p S_p)}{\partial t} + \frac{1}{|V_i|} \sum_{l \in \mathrm{adj}(i)} F_{p,il} = 0, \tag{1.2.1}$$

where $|V_i|$ is the volume of the i-th gridblock, and $F_{p,il}$ is the numerical flux function of phase p from cell i to cell l:

$$F_{p,il} = -|\partial V_{il}| K_{il} \rho_{p,il} \lambda_p(S_{il}) \left(\frac{(p_{p,l} - p_{p,i})(\mathbf{x}_l - \mathbf{x}_i)}{|\mathbf{x}_l - \mathbf{x}_i|^2} - \frac{\gamma_{p,il}(\mathbf{z}_l - \mathbf{z}_i)}{|\mathbf{x}_l - \mathbf{x}_i|} \right) \cdot \boldsymbol{\nu}_{il}, \tag{1.2.2}$$

where $|\partial V_{il}|$ is the area of the interface between cells i and l, $\mathbf{x}_i$ is the location of the center of cell i, $\mathbf{z}_i$ is the component of $\mathbf{x}_i$ along the direction of gravity, and $\boldsymbol{\nu}_{il}$ is the unit normal to the cell interface, pointing from cell i to cell l. The above discretization uses a two-point flux approximation, and we restrict ourselves to the two-point flux case in this dissertation. One should note, however, that multipoint flux approximations are also used occasionally in reservoir simulation, especially for tensorial permeability fields [2, 47, 48].

The literature on finite volume methods for multiphase flow is vast [87, 79, 16], and [6] describes the method in detail for various flow configurations. On the other hand, the use of finite-element methods for general-purpose simulation in industry is rare. Finite element methods are more flexible in terms of the treatment of unstructured grids, irregular boundaries, as well as anisotropic or tensorial permeability fields. As a result, there is active interest in using finite-element methods to develop finite-volume discretizations [53]. In this thesis, we restrict our discussion to finite volume methods, but the reader is referred to [1, 43, 86, 31] for more detailed discussion on finite element methods.

A peculiar feature of the spatial discretization used in reservoir simulation is the upstream weighting of saturation-dependent terms. Buoyancy and capillary forces may induce sonic points to the hyperbolic flux function (see Figure 2.1), but the exact location of the sonic point is a strong function of the total velocity and permeability, so it would be inconvenient to locate the sonic point for every cell interface. In practical simulations, the upstream direction for phase p is determined by the *potential gradient* of phase p. Since different phases can have different upstream directions, the resulting numerical flux functions are in fact a combination of mobilities, each evaluated at a different saturation. It can be shown [13] that these numerical flux functions are different from those used in classical CFD, such as the Godunov and Engquist-Osher schemes. In Chapter 2, we will study this upstream weighting in detail and discuss its convergence to the analytical solution under grid refinement.

1.2.2 Temporal discretization

A variety of temporal discretizations are commonly used in black-oil simulation. The most commonly used methods are:

1. *Implicit pressure, explicit saturation* (IMPES): All saturation-dependent coefficients in the flux terms are evaluated at the beginning of the time step ($t = t^n$), and pressure-dependent terms are evaluated at the end of the time step ($t = t^{n+1}$). In algorithmic terms, this amounts to (1) solving the pressure equation (1.1.18) for p^{n+1}, (2) computing the phase velocities $\mathbf{u}_p$ at (S^n, p^{n+1}), and

(3) updating the saturations using the mass-balance equations (1.1.7), (1.1.8) and a forward difference approximation for $\partial/\partial t$. Because of the explicit treatment of saturation, IMPES is only conditionally stable; the CFL condition for a 1D two-phase incompressible oil-water problem without gravity is given by (cf. [20])

$$\Delta t < \frac{\phi}{\dfrac{2K\lambda_w\lambda_o\,|dP_{cow}/dS_w|}{(\lambda_w+\lambda_o)\Delta x^2} + \dfrac{v_T df_w/dS_w}{\Delta x}}, \tag{1.2.3}$$

where v_T is the total velocity of the oil and water phases, and f_w is the fractional flow of the water phase:

$$f_w = \frac{\lambda_w}{\lambda_w+\lambda_o}\left[1 + \frac{K\lambda_o}{v_T}\frac{\partial P_{cow}}{\partial x}\right].$$

In the absence of capillarity, (1.2.3) reduces to the familiar CFL condition for the hyperbolic conservation law

$$\phi S_t + (v_T f_w(S))_x = 0.$$

Thus, Δt is $O(\Delta x^2)$ when capillarity is present, and $O(\Delta x)$ otherwise.

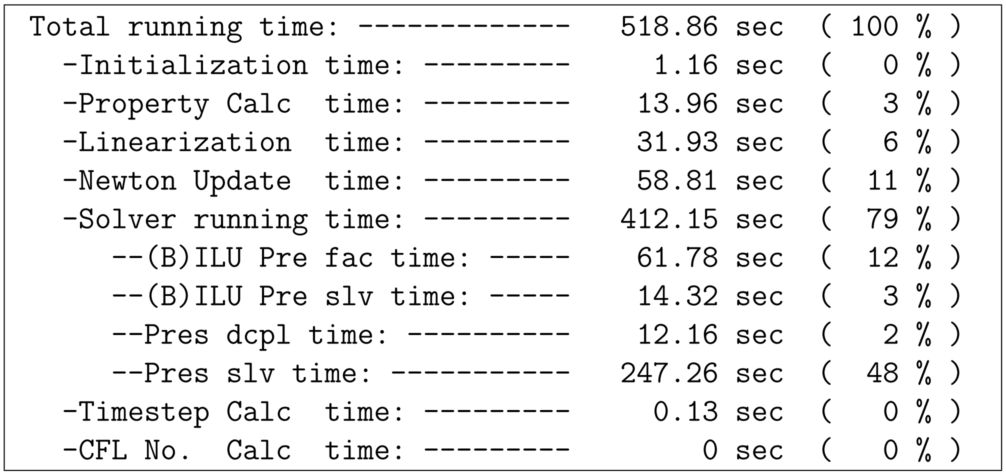
2. *Sequential implicit method* (SEQ): The sequential implicit method computes the new pressure p^{n+1} in exactly the same manner as IMPES, but it updates the saturations by solving the transport problem with implicit time-stepping [72, 82]. This amounts to an operator splitting method, in which the flow problem (resolution of the global pressure field) and the transport problem (advection of individual phases) are decoupled and solved sequentially. A more detailed description is given in Section 2.2.2. Since the transport problem is solved implicitly using a frozen total velocity field $\mathbf{v}_T$, SEQ is stable for any time-step size as long as $\mathbf{v}_T$ is conservative. However, for compressible flow, mass is generally not conserved for one of the phases; the mass-balance errors are proportional to the areal variation of ρ_o/ρ_w [6, 21] and can be significant for large time steps.

3. *Adaptive implicit method* (AIM): This method changes the level of implicitness adaptively for each cell, depending on the CFL limit for that cell. For a cell experiencing fast flows (i.e., the local CFL number is greater than 1), both the saturation and pressure are taken implicitly; if, on the other hand, the local CFL number is less than 1, the saturations are taken explicitly, whereas pressure is taken implicitly. More detailed descriptions and analyses can be found in [76, 36, 67, 26].

4. *Fully implicit method* (FIM): Both saturation and pressure variables are taken implicitly in every cell. A linear stability analysis [6], together with a more refined analysis based on linearized mobilities [61], strongly indicate (but do not provide a rigorous proof) that this method is unconditionally stable. However, it is also generally the most diffusive of the above mentioned schemes.

These methods differ in the level of implicitness of the saturation-dependent quantities, with IMPES having the least degree of implicitness and FIM having the most. Note that pressure is treated implicitly in all methods. This is because the pressure equation is either weakly parabolic (and nearly elliptic) in the compressible case, or elliptic in the incompressible case. Hence, in the compressible case, explicit pressure treatment would entail a time-step restriction proportional to Δx^2, which is unacceptably severe. In the incompressible case, the pressure equation degenerates into a constraint that is required to ensure global conservation, which must be satisfied by the numerical solution. Thus, it is also necessary to treat pressure implicitly in the incompressible case.

Clearly, a method with a lower level of implicitness would incur a lower computational cost per time step. However, the difference in computational cost between explicit and implicit methods (such as IMPES and FIM) is not as pronounced as one would expect, since the "explicit" IMPES still needs to solve an implicit pressure equation at every time step. Figure 1.1 shows the amount of time the simulator spends in each module during a typical black-oil simulation when FIM is used. Even for FIM, the pressure solve represents almost half of the total running time, and about 60% of the solver time. So in this case, IMPES would be faster than FIM only if the FM time step is chosen such that the maximum CFL number is less than



```
Total running time: -------------   518.86 sec  ( 100 % )
  -Initialization time: ---------     1.16 sec  (   0 % )
  -Property Calc  time: ---------    13.96 sec  (   3 % )
  -Linearization  time: ---------    31.93 sec  (   6 % )
  -Newton Update  time: ---------    58.81 sec  (  11 % )
  -Solver running time: ---------   412.15 sec  (  79 % )
     --(B)ILU Pre fac time: -----    61.78 sec  (  12 % )
     --(B)ILU Pre slv time: -----    14.32 sec  (   3 % )
     --Pres dcpl time: ----------    12.16 sec  (   2 % )
     --Pres slv time: -----------   247.26 sec  (  48 % )
  -Timestep Calc  time: ---------     0.13 sec  (   0 % )
  -CFL No.  Calc  time: ---------        0 sec  (   0 % )
```

Figure 1.1: Timing report for a typical black-oil simulation run. The above simulation is performed on a 3D, two-phase heterogeneous model with 141900 grid blocks.

1.67. In practice, reasonable time steps yield maximum CFL numbers that are much larger than 1 because of the presence of sources and sinks, as well as spatial variations in permeability and porosity. However, the impact of these high CFL numbers on overall accuracy is minimal because they only occur in a few cells. Figure 1.2 shows the saturation profiles for the FIM and IMPES solutions in a 2D water flood problem. The maximum saturation difference between the two solutions is 0.036, which is negligible considering the uncertainty in the reservoir characterization. In this case, FIM takes only 113 time steps to reach T_{final}, whereas IMPES takes 1318 steps, so FIM is clearly more efficient.

The above example, in which the high CFL numbers do not significantly affect solution accuracy, is typical among reservoir models of practical interest. Such models are generally highly heterogeneous with permeability variations up to several orders of magnitude. Moreover, wells can be completed anywhere in the reservoir model and can operate in a wide variety of ways, often resulting in CFL limits that are unacceptably severe. Thus, reservoir simulators typically use implicit time-stepping for robustness and efficiency. Consequently, efficient linear and nonlinear solvers for the fully-implicit problem can be the crucial factor in determining the efficiency of reservoir simulators.

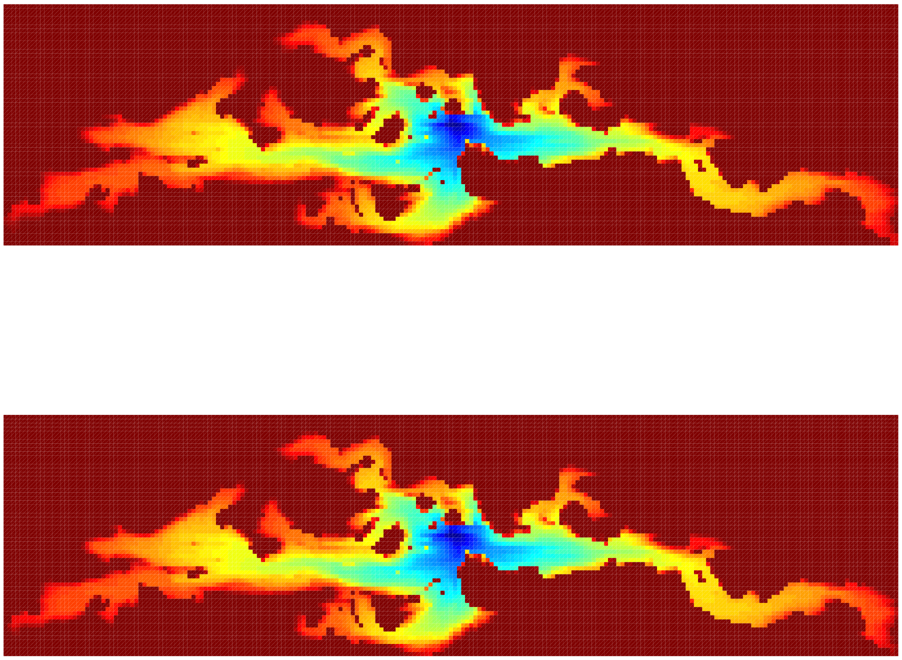


Figure 1.2: A comparison between FIM (top) and IMPES (bottom) saturation profiles for a 2D heterogeneous reservoir. The permeability and porosity fields are taken from the 51st layer of the SPE 10 reservoir [19].

Higher-order methods for reservoir simulation have been an active area of research in recent years. With the exception of streamline methods, which can take advantage of high-order 1D integrators readily [52], higher-order methods are still primarily in the development stage and are not yet routinely used in commercial simulators. A major impediment to the wide-spread adoption of higher-order methods is the loss of positivity, which leads to spurious oscillations as the initial profile is integrated forward in time. An important result due to Bolley and Crouzeix [12] states that a method that preserves positivity for all Δt is at most first-order accurate. An elaborate discussion on higher-order methods is beyond the scope of this thesis; see [9, 18, 11, 26, 77] for a detailed discussion.

1.2.3 Solution of nonlinear equations

Since all temporal discretizations contain some level of implicitness, the simulator needs to solve a large system of nonlinear algebraic equations at each time step. The size and properties of this system, of course, depend on the number and nature of the implicit variables. For IMPES, the nonlinear system will by N-by-N, where N is the number of grid blocks (control volumes) in the domain, and the equations will inherit the parabolic/elliptic nature of the pressure equation. For FIM, on the other hand, we would have an n_pN-by-n_pN system, where n_p is the number of fluid phases, and the equations would be of mixed hyperbolic-parabolic type. As a result, the bulk of the simulation time (80% to 90%, cf. Figure 1.1) is spent on solving these large systems. It is therefore crucial, for the sake of efficiency and robustness, that the linear and nonlinear solvers exploit the structure and properties of these discrete equations.

Nonlinear solvers

The most commonly used nonlinear solvers in reservoir simulation are all variations on the basic Newton method:

$$\begin{aligned} &\text{Solve} \quad J(x^{(\nu)})\,\delta x^{(\nu)} = -R(x^{(\nu)}) \quad \text{for } \delta x^{(\nu)}, \\ &\text{Set} \quad\;\; x^{(\nu+1)} = x^{(\nu)} + \delta x^{(\nu)}, \quad \nu = 0, 1, 2, \dots, \end{aligned} \tag{1.2.4}$$

where $R(x)$ is the residual function and $J(x) = \partial R/\partial x$ is the Jacobian matrix. Newton's method is popular because of its local quadratic convergence and its general applicability. For residual functions arising from discretized PDEs, the resulting Jacobian is generally sparse and structured, which means the linear systems can be solved efficiently. Also, quadratic convergence means Newton's method is very fast when good initial guesses are available. For time-dependent problems, a natural initial guess is the saturation and pressure profiles from the previous time step. Assuming the profiles vary continuously with time (which is always true for pressure, and true for saturations away from shock fronts), the old time-step values will be close to the solution provided Δt is small enough. However, when the time step is too large, it is possible for Newton's method to diverge, since the residual functions are in general non-convex and possibly non-monotonic (see Figure 2.1). When faced with non-convergence, the simplest approach is to cut the time-step size and rerun Newton's method with the smaller time step. Such time-step cuts are very expensive, since they mean we must throw away the results of all previous iterations and start over. Thus, one should avoid time-step cuts as much as possible.

One way to avoid time-step cuts is to take small enough time-steps. However, in practice, one does not want to choose time-step sizes based on the nonlinear solver for the following reasons:

1. The use of excessively small time steps reduces the benefits of using FIM, since we would not be taking advantage of its ability to handle long time steps;

2. It is difficult to decide whether Newton's method would converge for a particular time-step size without actually performing the iterations;

3. The time-step size should be chosen based on the desired solution accuracy (e.g., bounds on numerical diffusion errors) instead of the ability of the nonlinear solver to converge a time step.

For these reasons, several modifications to the basic Newton's method have been proposed to ensure global convergence, or at least to enlarge the region of convergence to the point that the algorithm will converge for all Δt of practical interest. Globalization techniques for general nonlinear residual functions, such as line search and trust-region methods, are discussed in [29]. In our experience, line search methods, in which the search direction is scaled by a single step-length parameter α, are inadequate for reservoir simulation problems because (1) the residual norm is sensitive to diagonal scaling, and the correct scaling for the phase conservation equations is not obvious in most problems; (2) α is often very small when flow reversal due to gravity occurs across several cell interfaces; (3) a number of backtracking steps is often needed to guarantee a sufficient decrease in the residual, and function evaluations are quite expensive, since each evaluation involves calculating fluid properties and pressure gradients for every cell in the domain.

Another method, which is implemented in the commercial simulator Eclipse, is the so-called Appleyard chop [37]. It limits, on a cell-by-cell basis, the allowable saturation and pressure changes within a nonlinear iteration to a fixed (but empirically determined) threshold. When the threshold parameters are chosen properly, the method is quite robust and the number of time-step cuts is often small. However, because large saturation changes are disallowed, the method can lead to unnecessarily slow convergence, especially in cases where Newton's method actually works well (such as problems with convex fractional flow functions).

Other methods for solving general nonlinear systems (e.g., continuation methods) can be found in [29, 59]. Such methods, however, are not used in general-purpose simulation in industry.

Linear solvers

To solve the linear system (1.2.4), early reservoir simulators [51, 62, 6] used either direct methods (Gaussian elimination) or stationary iterative methods such as successive over-relaxation (SOR), alternating direction implicit method (ADI), or Stone's strongly implicit procedure (SIP) [73]. With the advent of Krylov accelerators such as ORTHOMIN [80] and GMRES [69], iterative methods became more popular, and the need for efficient preconditioning techniques has increased. In addition to preconditioners derived from stationary methods, other preconditioners have been developed by the reservoir simulation community to handle the linear equations arising from fully-implicit simulation. Examples include:

1. *Incomplete factorization* (ILU): Originally developed by Watts [84] to handle Jacobian matrices from structured grids, this technique has been generalized to handle other sparse matrices. For a thorough discussion of ILU and its variants, see [68].

2. *Nested factorization*: This method was introduced by Appleyard *et al.* in [5] and subsequently improved by Appleyard and Cheshire in [3]. It exploits the band structure in three-dimensional problems to produce an approximate factorization $M = LU$, such that the error matrix $E = M - A$ has zero column sums. In physical terms, this means global mass balance is preserved by the approximate factors, yielding a better preconditioner than ILU.

3. *Constrained pressure residual* (CPR): Proposed by Wallis *et al.* [81], CPR is a two-stage preconditioner in which the residual vector is constrained to lie in some subspace V via a projection process. The choice of constraint subspace determines the effectiveness of the preconditioner. With the emergence of fast elliptic solvers such as algebraic multigrid [74], CPR has become one of the most attractive preconditioners for reservoir simulation problems [17].

Behie [8] provides a comparison among the three preconditioners above. In Chapter 5, the spectral properties of CPR-preconditioned Jacobians are discussed in detail.

1.3 Thesis outline

In this thesis, we make two contributions to the existing literature on reservoir simulation. On the algorithmic side, we present a new ordering scheme for the equations and unknowns for the discrete mass-balance equations (1.2.1), (1.2.2). This new ordering exploits flow direction information and allows us to derive a more efficient nonlinear solver as well as an improved linear preconditioner. On the theoretical side, we present a rigorous nonlinear analysis of phase-based upstream discretization. We show that the discretization yields a well-defined, stable and monotonic method that converges to the entropy solution for arbitrary CFL numbers. This complements the existing literature [6, 61] in which only stability is established using a linear or linearized stability analysis.

In Chapter 2, we analyze phase-based upstreaming in detail. We show how the FIM formulation in 1D, as well as SEQ in multiple dimensions, can be cast as a monotone implicit scheme. We then extend the work of Rheinboldt on M-functions and Gauss-Seidel iterations [64] to show that the discretized equations always have a unique solution, which can be found using the nonlinear Gauss-Seidel process. We also show that the discrete solution converges to the entropy solution under grid refinement, and we investigate the accuracy of the discrete solutions for different time-step sizes and spatial grids. This chapter is of a more theoretical nature, and practitioners of reservoir engineering who are familiar with the discretizations can go directly to Chapter 3 for a more algorithms-related discussion.

In Chapter 3, we introduce phase-based potential ordering, which reorders the equations and variables in the nonlinear system in a way that exploits flow direction information and eventually allows a partial decoupling of the problem into a sequence of single-cell problems that are easy to solve. This ordering is valid for both two-phase and three-phase flow, and it can handle countercurrent flow due to gravity and/or capillarity.

In Chapter 4, we propose a reduced-order Newton algorithm, which makes use of the phase-based potential ordering in Chapter 3 to reduce the size of the nonlinear system. The latter is then solved using Newton's method. We analyze its convergence behavior for 1D cocurrent problems, and we show a variety of examples (two- and three-phase flow, with and without gravity) illustrating its effectiveness in dealing with large, complex heterogeneous problems.

In Chapter 5, we analyze the two-stage CPR preconditioner in detail and propose an improved second-stage preconditioner that uses a cell-based potential ordering. This approach reduces the sensitivity of CPR to flow configurations, and this reduction in sensitivity is both justified theoretically and observed from numerical experiments. We also experiment with directly preconditioning the Schur complement problem that arises from the phase-based potential order reduction.

We present our conclusions and outline future directions in Chapter 6.

Chapter 2

Analysis of Upstream Weighting

2.1 Background

As mentioned in Chapter 1, the multiphase flow equations give rise to a system of n conservation laws (where n is the number of immiscible fluid phases), defined over $\Omega \subset \mathbb{R}^k$ $(1 \leq k \leq 3)$, each of the form

$$\frac{\partial(\phi \rho_j S_j)}{\partial t} + \nabla \cdot (\rho_j u_j) = \rho_j q_j, \qquad j = 1, \ldots, n, \tag{2.1.1}$$

and generalized Darcy's law

$$u_j = -K\lambda_j \nabla(p_j - \gamma_j z), \tag{2.1.2}$$

where $\phi = \phi(x)$ is the porosity of the medium (with $0 < \phi \leq 1$), $K = K(x) > 0$ is the absolute permeability, $z = z(x)$ is the depth variable; and for each phase $j = 1, \ldots, n$, ρ_j is the density, S_j is the saturation (i.e. the volume fraction occupied by phase j), u_j is the volumetric flux vector, q_j is the source or sink term, $\lambda_j = \lambda_j(S_1, \ldots, S_n)$ is the phase mobility, p_j is the pressure, and γ_j is the gravitational force. In addition, we have the algebraic relations:

$$\text{Saturation constraint:} \qquad \sum S_j = 1, \tag{2.1.3}$$

$$\text{Capillary pressure constraint:} \quad p_j - p_{j+1} = P_{cj}(S_1, \ldots, S_n), \quad j = 1, \ldots, n-1. \tag{2.1.4}$$

The above system of PDEs exhibits a mixed hyperbolic-parabolic character, which becomes apparent when we consider the various limiting cases. If we assume constant densities and neglect capillary pressure relations (i.e. we assume $p_1 = \cdots = p_n \equiv p$), then we can sum (2.1.1) over $j = 1, \ldots, n$ and invoke the saturation constraint to get

$$-\nabla \cdot \Big(K\lambda_T \nabla p - K\nabla z {\textstyle\sum_j} \gamma_j \lambda_j \Big) = {\textstyle\sum_j} q_j, \tag{2.1.5}$$

where $\lambda_T = \sum_j \lambda_j$ is the total mobility. Thus, for a given saturation distribution, the pressure field satisfies an elliptic PDE. On the other hand, when the total velocity $u_T = \sum_j u_j$ is constant over the domain (which is the case for flow in a one-dimensional porous medium), we can rewrite u_j as

$$u_j = \frac{\lambda_j}{\lambda_T} \left(u_T - K\nabla z {\textstyle\sum_l} \lambda_l (\gamma_l - \gamma_j) \right), \tag{2.1.6}$$

which is a function of the saturations $S_1, \ldots, S_n$ only. Thus, if we substitute (2.1.6) into (2.1.1), we get

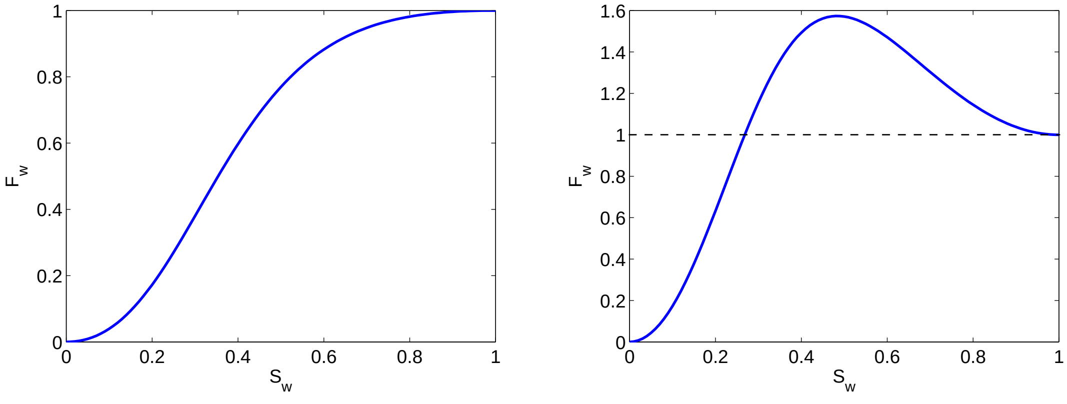
$$\phi \frac{\partial S_j}{\partial t} + \nabla \cdot u_j(x, S_1, \ldots, S_n) = 0, \qquad j = 1, \ldots, n-1. \tag{2.1.7}$$

This means saturation behaves like the solution to a system of first-order hyperbolic PDEs, so one should expect discontinuous saturation profiles. In higher dimensions, there is generally a strong coupling between pressure and saturation, due to the saturation dependence of λ_j and λ_T in (2.1.5) and the dependence of u_T on the pressure field in (2.1.6). In addition, the porosity ϕ and permeability K are highly oscillatory, non-smooth functions of x, and $K(x)$ can vary by several orders of magnitude over the domain Ω. The large variability of ϕ and K leads to local CFL limits that are unacceptably severe when explicit schemes are used. As a result, the discretization of choice for most reservoir simulators is the *fully-implicit method* (FIM), which uses finite volume in space and backward Euler in time. The numerical flux functions, which approximate the u_j as defined in (2.1.2), use a two-point finite difference to approximate ∇p and *phase-based upstream weighting* to approximate $\lambda_j(S)$. In other words, to approximate u_j at the interface of cells a and b (centered at x_a and x_b), we evaluate $\lambda_j(S)$ at

$$S = \begin{cases} S(x_a) & \text{if } -\nabla(p_j - \gamma_j z) \cdot \nu_{ab} \geq 0, \\ S(x_b) & \text{otherwise,} \end{cases} \tag{2.1.8}$$

where ν_{ab} is the unit vector normal to the interface, pointing from a to b. The resulting numerical flux functions are different from those used in classical CFD, such as the Godunov and Engquist-Osher schemes [13]. Despite being only first-order accurate, phase-based upstreaming is the preferred upwind method in reservoir simulation because it is physically intuitive, and because it is generally easier to verify a consistency condition such as (2.1.8) than to identify potential sonic points, which vary over the domain and are strong functions of permeability and total velocity. This is especially true for the fully-implicit method because the total velocity at time t^{n+1} is usually unknown.

Note that in (2.1.8) it is possible for $-\nabla(p_j - \gamma_j z) \cdot \nu_{ab}$ to have different signs for different j, meaning the upstream directions can be different for different phases when buoyancy forces are significant; this is known as *countercurrent flow* in reservoir engineering literature. In one-dimensional porous media, countercurrent flow manifests itself through the presence of sonic points in the flux function u_j; thus, the flux function for a countercurrent flow problem would typically look like the one shown in Figure 2.1(b), whereas without countercurrent flow it would look more like Figure 2.1(a). A detailed treatment of phase-based upstreaming is given in [13], in which the authors showed that, when explicit time-stepping is used on a two-phase flow problem, phase-based upstreaming leads to a monotone difference scheme, as long as the appropriate CFL condition is satisfied. This in turn implies that the solution of



(a) (b)

Figure 2.1: Flux functions for 1D incompressible two-phase flow: (a) Co-current flow (no buoyancy effects), (b) Countercurrent flow due to gravity.

the explicit schemes converge to the entropy solution of the two-phase equations

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = 0, \tag{2.1.9}$$

$$f(S) = u_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2}\left[u_T + K\lambda_2(\gamma_1 - \gamma_2)\frac{\partial z}{\partial x}\right] \tag{2.1.10}$$

as $\Delta t, \Delta x \to 0$ while satisfying the CFL condition. The goal of this chapter is to extend this result to the fully-implicit case. This leads us to study the more general problem of implicit monotone schemes, which would then include the multiphase flow problem as a special case.

The use of implicit time-stepping leads to a (typically large) system of nonlinear algebraic equations that must be solved for each time step. Moreover, the residual functions are generally non-differentiable because of upstreaming criteria of the form (2.1.8); thus, the existence of a unique solution to these systems of equations is not immediately obvious. For implicit monotone schemes for 1D scalar conservation laws, Lucier [50] showed that a solution to the discrete problem exists and is unique whenever the initial data is bounded and has bounded total variation. The proof of existence, which relies heavily on Crandall-Liggett theory [23], proceeds along the following lines (see [27, Chapter 3] for more details). First, one shows that the residual function R for the numerical scheme defines an m-accretive operator in the L^1-norm. Then by the Crandall-Ligett theorem, the ODE

$$\frac{du}{dt} = -Ru, \quad u(0) = x \tag{2.1.11}$$

has a unique solution for $t \in [0, \infty)$ for any initial point x. Let $u(t; x)$ denote the solution of (2.1.11) with starting point x. Then one shows that the *Poincaré operator* P_ω, which maps the point x to the point $u(\omega, x)$, is strictly contractive. Then by Banach's fixed point theorem, P_ω has a unique fixed point x_0. One then proceeds to prove that $u(t; x_0) = x_0$ for all $0 \leq t \leq \omega$; thus, $du/dt = 0$, which implies $Rx_0 = 0$.

While this argument does prove the existence and uniqueness of a solution to the discretized problem, the proof does not suggest a practical algorithm for finding the solution. In section 3, we present an alternate constructive proof of existence by showing that the classical Gauss-Seidel and Jacobi iterations converge for this class of problems. In fact, we show that the iterative methods converge whenever the initial data for the discrete problem is bounded, so the implicit scheme is well-defined even when the initial data does not have bounded variation in $\mathbb{R}$. The well-definedness of the numerical scheme, together with the total variation diminishing (TVD) property and the existence of a discrete entropy inequality, imply that the numerical scheme converges to the entropy solution as the mesh is refined (i.e., as $\Delta x \to 0$). This result holds for any mesh ratio $\lambda = \Delta t/\Delta x$ (i.e., for any Courant number).

2.2 Two model problems

In this section, we present two model problems from porous media flow, both of which contain a hyperbolic subproblem that can be analyzed using the theory developed in this chapter.

2.2.1 FIM for 1D problem with gravity

Consider a one-dimensional model problem with:

- incompressible two-phase flow,
- zero capillarity ($p_w = p_o \equiv p$),
- an injection boundary condition on the left, and
- a pressure boundary condition on the right.

In this case, the continuous problem (2.1.1)–(2.1.2) can be rewritten as

$$\phi(x)\frac{\partial S_p(x)}{\partial t} + \frac{\partial u_p(x)}{\partial x} = 0, \quad x_L < x < x_R, \tag{2.2.1}$$

$$u_p(x) = -K(x)\lambda_p(S_w(x))\left(\frac{dp}{dx} - \gamma_p\frac{dz}{dx}\right), \tag{2.2.2}$$

for $p = w, o$ (water and oil), together with the saturation constraint $S_o + S_w = 1$, the initial condition $S_w(x, 0) = S^0(x)$ for $x \in [x_L, x_R]$, and boundary conditions

$$u_p(x_L) = q_{p,L}, \qquad p(x_R) = p_R.$$

We assume that the injection velocities $q_{w,L}$ and $q_{o,L}$ are non-negative, and that the total velocity $q_{T,L} := q_{w,L} + q_{o,L}$ is strictly positive. (These assumptions cover the most interesting cases, such as oil recovering by water-flooding.) This formulation, which contains pressure variables, is known as the *parabolic form* of the problem, since it represents the incompressible limit of a parabolic problem. We can also derive the *hyperbolic* or "fractional flow" form of the problem by eliminating the pressure variables as follows. The discretized PDEs can be written as

$$\frac{\phi_i(S_{w,i} - S_{w,i}^{old})}{\Delta t} + \frac{F_{w,i+1/2} - F_{w,i-1/2}}{\Delta x} = 0, \tag{2.2.3a}$$

$$\frac{\phi_i(S_{w,i}^{old} - S_{w,i})}{\Delta t} + \frac{F_{o,i+1/2} - F_{o,i-1/2}}{\Delta x} = 0, \tag{2.2.3b}$$

where

$$F_{p,i+1/2} = K_{i+1/2}\lambda_{p,i+1/2}\left(\frac{p_i - p_{i+1}}{\Delta x} + g_p\right), \qquad p = o, w, \tag{2.2.4}$$

with $g_p = \gamma_p \Delta z/\Delta x$, $i = 1, \ldots, N$. The numerical boundary conditions become

$$F_{p,1/2} = q_p, \quad p = o, w, \tag{2.2.5}$$

$$p_{N+1} = 2p_R - p_N. \tag{2.2.6}$$

For the remainder of this section, we assume without loss of generality that $g_w \geq g_o$; in the case of $g_w < g_o$, the same argument would hold by considering the oil phase instead of the water phase. To eliminate the pressure variables p_i, first note that summing equations (2.2.3a) and (2.2.3b) and rearranging gives

$$F_{w,i+1/2} + F_{o,i+1/2} = F_{w,i-1/2} + F_{o,i-1/2} = q_w + q_o =: q_T.$$

In other words, the total flux is constant across any interface, and this flux is denoted by q_T, which is equal to $q_{T,L}$. Summing Equation (2.2.4) through $p = o, w$, we can express the pressure gradient $(p_i - p_{i+1})/\Delta x$ in terms of q_T:

$$q_T = K_{i+1/2}\left[\lambda_{T,i+1/2}\frac{p_i - p_{i+1}}{\Delta x} + (\lambda_{w,i+1/2}\, g_w + \lambda_{o,i+1/2}\, g_o)\right],$$

where $\lambda_{T,i+1/2} = \lambda_{w,i+1/2} + \lambda_{o,i+1/2}$. Thus,

$$\frac{p_i - p_{i+1}}{\Delta x} = \frac{q_T - K_{i+1/2}(\lambda_{w,i+1/2}\, g_w + \lambda_{o,i+1/2}\, g_o)}{K_{i+1/2}(\lambda_{w,i+1/2} + \lambda_{o,i+1/2})}. \tag{2.2.7}$$

Substituting into (2.2.4) for the water phase gives

$$\begin{aligned} F_{w,i+1/2} &= \frac{\lambda_{w,i+1/2}}{\lambda_{T,i+1/2}}\left[q_T + K_{i+1/2}\lambda_{o,i+1/2}\Delta g\right] \\ &= F_{w,i+1/2}(S_{w,i}, S_{w,i+1}), \end{aligned} \tag{2.2.8}$$

where $\Delta g = g_w - g_o \geq 0$. This, together with (2.2.3a):

$$\phi_i(S_{w,i} - S_{w,i}^{old}) + \frac{\Delta t}{\Delta x}(F_{w,i+1/2} - F_{w,i-1/2}) = 0, \tag{2.2.9}$$

leads to a numerical scheme with exactly the same form as (2.3.1), except for the boundary conditions. Clearly, the treatment of boundary conditions will significantly affect the stability and accuracy of the numerical scheme. However, in order to understand the behavior of the numerical scheme at interior points, we will replace the initial-boundary value problem (2.2.1) with an initial value problem on an infinite domain with appropriate initial conditions. In particular, we replace the injection boundary condition with

$$S^0(x) = f^{-1}(q_{w,L}/q_{T,L}), \qquad x < x_L, \tag{2.2.10}$$

and the pressure boundary condition with

$$S^0(x) = S^0(x_R), \qquad x > x_R. \tag{2.2.11}$$

The modified continuous problem will yield a solution identical to (2.2.1) for $0 < t < T_{BT}$, where T_{BT} is the breakthrough time (i.e. the time at which the shock wave arrives at the pressure boundary). Note that since f is one-to-one over the interval $I = \{S : 0 \le f(S) < 1\}$ (see Figure 2.1), and since $q_{w,L} \le q_{T,L}$ by assumption, (2.2.10) is well-defined unless $q_{o,L} = 0$. (If $q_{o,L} = 0$, we define $u_0(x) = \inf f^{-1}(1)$, where f^{-1} denotes the inverse image.)

Phase-based upstreaming

Recall from section 2.1 (cf. Equation (2.1.8)) that the mobilities $\lambda_{p,i+1/2}$ are evaluated using the upstream saturations with respect to the flow direction of phase p:

$$\lambda_{p,i+1/2} = \begin{cases} \lambda_p(S_i) & \text{if } \frac{1}{\Delta x}(p_i - p_{i+1}) + g_p \ge 0, \\ \lambda_p(S_{i+1}) & \text{otherwise.} \end{cases} \tag{2.2.12}$$

In light of (2.2.7), we can rewrite the upstream conditions as

$$\lambda_{p,i+1/2} = \begin{cases} \lambda_p(S_i) & \text{if } q_T + K_{i+1/2}(g_p - g_q)\lambda_{q,i+1/2} \ge 0, \\ \lambda_p(S_{i+1}) & \text{otherwise,} \end{cases} \tag{2.2.13}$$

where the subscript q denotes the phase other than phase p. Even though pressure dependence has been eliminated, Equation (2.2.13) still does not explicitly define the upstream direction for λ_p, since the definition of upstream is in terms of the (yet undetermined) mobility of the other phase $\lambda_{q,i+1/2}$. For explicit numerical schemes, Brenier and Jaffré have shown in [13] how to explicitly determine the upstream direction for each phase for a given saturation profile $\{S_i^n\}$. In the special case of two-phase flow, they define the following quantities:

$$\begin{aligned}\theta_{o,i+1/2} &= q_T - K_{i+1/2}\Delta g \lambda_w(S_i^n),\\ \theta_{w,i+1/2} &= q_T + K_{i+1/2}\Delta g \lambda_o(S_{i+1}^n).\end{aligned}$$

These quantities correspond precisely to the condition in (2.2.13), but the condition is evaluated at S_i^n for θ_o and S_{i+1}^n for θ_w. Clearly $\theta_{w,i+1/2} > 0$, since $\Delta g \geq 0$. The correct upstream directions are then given by

$$\begin{aligned}&\lambda_{o,i+1/2}^n = \lambda_o(S_i^n), &&\lambda_{w,i+1/2}^n = \lambda_w(S_i^n) &&\text{if } 0 \leq \theta_{o,i+1/2} \leq \theta_{w,i+1/2},\\ &\lambda_{o,i+1/2}^n = \lambda_o(S_{i+1}^n), &&\lambda_{w,i+1/2}^n = \lambda_w(S_i^n) &&\text{if } \theta_{o,i+1/2} \leq 0 \leq \theta_{w,i+1/2}.\end{aligned}$$

Thus, for an explicit time-marching scheme, the numerical fluxes are completely defined by these conditions, and there is no need to go back to the original definition (2.2.12) involving unknown pressure values. However, this is not the case for an implicit time-marching scheme (such as backward Euler), since the upstream directions must be consistent with the saturation values *at the end of the time step*, i.e. with the saturation profile $\{S_i^{n+1}\}$. Because of this consistency requirement, it is not clear a priori that a solution to the parabolic form of the problem (2.2.3) even exists. Our approach to proving that a solution exists is to rely on the hyperbolic form g(2.2.8)–(2.2.11). From the above derivation, it is evident that if $\{(S_i, p_i)\}_{i=1}^N$ is any solution to the parabolic form (2.2.3)–(2.2.6), then $\{S_i\}_{i=1}^N$ must be a solution to the hyperbolic problem. Thus, the key idea is to begin by finding the correct saturation profile $\{S_i\}$ via (2.2.8)–(2.2.11), with a numerical flux that automatically ensures consistency with the upstream directions; once the $\{S_i\}$ are known, we can easily solve for the pressure part because the pressure equation is linear. We distinguish two cases:

1. If $K_{i+1/2}\Delta g \lambda_{w,max} \le q_T$, then $\theta_{o,i+1/2} \ge 0$ always, so we revert to a single-point upstream scheme $F_{i+1/2} = F_{i+1/2}(S_i)$;

2. If $K_{i+1/2}\Delta g \lambda_{w,max} > q_T$, then by the monotonicity of $\lambda_w(S)$, there exists a unique $0 < S_c < 1$ such that

$$K_{i+1/2}\Delta g \lambda_w(S_c) = q_T.$$

   Then the numerical flux, which is to be evaluated at time t^{n+1}, is defined as

$$F_{w,i+1/2}(S_i, S_{i+1}) = \begin{cases} \dfrac{\lambda_w(S_i)\left[q_T + K_{i+1/2}\lambda_o(S_i)\Delta g\right]}{\lambda_w(S_i) + \lambda_o(S_i)} & \text{if } 0 \le S_i \le S_c, \\ \dfrac{\lambda_w(S_i)\left[q_T + K_{i+1/2}\lambda_o(S_{i+1})\Delta g\right]}{\lambda_w(S_i) + \lambda_o(S_{i+1})} & \text{if } S_c < S_i \le 1. \end{cases} \tag{2.2.14}$$

A plot of the numerical flux $F_w(u, v)$ in the latter case is shown in Figure 2.2. The black curve on the surface, which shows the value of $F(u, v)$ along the line $u = v$, is identical to the continuous flux function in Figure 2.1(b). Thus, it is evident that the numerical flux satisfies the consistency condition $F(u, u) = f(u)$. Even though $f(u)$ itself is non-monotonic, the plot clearly shows that $F(u, v)$ is an increasing function of u and a decreasing function of v. This monotonicity property is what makes upstream weighting amenable to a Gauss-Seidel type analysis. Also notice that the numerical flux is independent of the downstream saturation v inside the cocurrent region ($0 \le u \le S_c \approx 0.27$), but becomes a function of both variables when $u > S_c$. Finally, $F(u, v)$ is Lipschitz continuous, but non-differentiable along the line $u = S_c$ because of the upstream condition (2.2.14). The following theorem, which summarizes several results by Brenier and Jaffré [13], shows that upstream-weighted fluxes generally satisfy the monotonicity property.

Theorem 2.1. *Assume that the mobility of phase p is increasing with the saturation of the same phase and decreasing with the saturation of the other phase, for $p = o, w$ (oil and water). Then the numerical fluxes obtained from phase-based upstreaming defined by* (2.2.8), (2.2.13) *are (1) Lipchitz continuous, (2) consistent with the continuous* *flux function (i.e., $F(u,u) = f(u)$), (3) non-decreasing with respect to $S_{w,i}$, and (4) non-increasing with respect to $S_{w,i+1}$.*

The hypothesis on phase mobilities is physically realistic [6]. These properties are sufficient to ensure that the hyperbolic problem with implicit time-stepping possesses a unique solution $\{S_i^{n+1}\}$, which must also be the correct saturation profile for the parabolic problem. To solve for pressure, we use Equation (2.2.7):

$$\frac{p_i - p_{i+1}}{\Delta x} = \frac{q_T - K_{i+1/2}(\lambda_{w,i+1/2}\, g_w + \lambda_{o,i+1/2}\, g_o)}{K_{i+1/2}(\lambda_{w,i+1/2} + \lambda_{o,i+1/2})}$$

for $i = 1, \dots, N$, and the boundary condition (2.2.6):

$$p_{N+1} = 2p_R - p_N.$$

Since $\{S_i^{n+1}\}$ is now known, the right-hand side of (2.2.7) also completely determined. Thus, the vector p of pressures actually satisfies $Ap = b$, where A is an $N \times N$ upper triangular matrix with a nonzero diagonal. So A is nonsingular, which means there is a unique pressure profile $\{p_i^{n+1}\}$ that satisfies (2.2.7) and (2.2.6). It is easy to see that this pressure profile is consistent with the upstream condition (2.2.12): because of (2.2.7), this upstream condition is equivalent to (2.2.13), and the conditions therein are precisely the ones we use to define the numerical flux function (2.2.14) for the hyperbolic problem. Hence, we have shown that the parabolic form (2.2.3)–(2.2.6) has a unique solution, given by the above $\{(S_i^{n+1}, p_i^{n+1})\}$.

2.2.2 SEQ for multidimensional problems

In multiple dimensions, it is no longer possible to eliminate pressure variables, because the total velocity u_T is generally a function of space and time. Thus, the system of PDEs (2.1.1)–(2.1.2) does not reduce to a purely hyperbolic problem, which means we cannot directly apply our existence and uniqueness results to the fully-implicit method in this case. Nonetheless, our analysis does apply to the *sequential-implicit method* (see section 1.2.2). In each time step in SEQ, we first solve the discrete version of the (linear) elliptic equation (2.1.5), in which the saturation-dependent coefficients

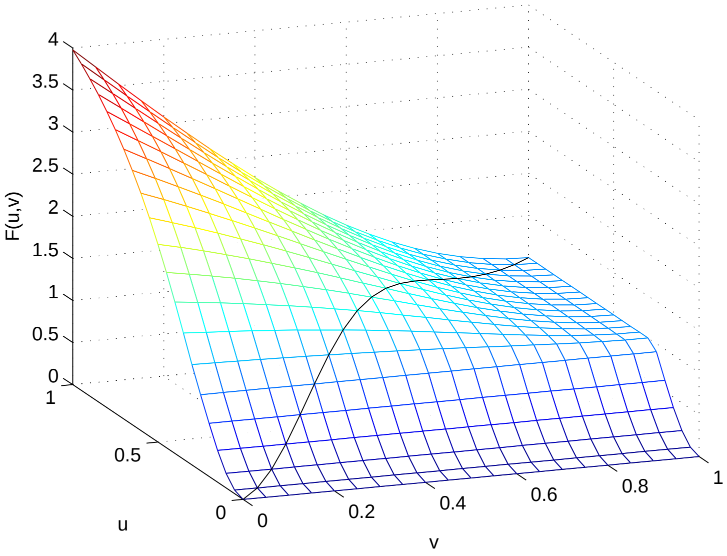


Figure 2.2: The numerical flux function $F(u, v)$ corresponding to the fractional flow in Figure 2.1(b). The black curve along the diagonal indicates the value of $F(u, u) = f(u)$.

are taken at time t^n. In other words, we solve for p^{n+1} via

$$-\nabla \cdot \left[K\lambda_T(S^n)\nabla p^{n+1} - K\nabla z \sum_j \gamma_j \lambda_j(S^n) \right] = \sum_j q_j. \tag{2.2.15}$$

Next, we compute the total velocity

$$u_T^* = \sum_j u_j^* = -\sum_j K\lambda_j(S^n)\nabla(p^{n+1} - \gamma_j z). \tag{2.2.16}$$

Finally, we compute the saturations S_j^{n+1} ($j = 1, \ldots, n-1$) by solving the discrete version of (2.1.6) and (2.1.7) with *implicit time-stepping*:

$$\begin{aligned} \phi \frac{\partial S_j}{\partial t} + \nabla \cdot u_j(x, S_1, \ldots, S_n) &= 0, \\ u_j &= \frac{\lambda_j}{\lambda_T} \left(u_T^* - K\nabla z {\textstyle\sum_l} \lambda_l(\gamma_l - \gamma_j) \right). \end{aligned}$$

Essentially, the SEQ method decouples the system into an elliptic and a hyperbolic subproblem. A finite-volume discretization of (2.1.6) and (2.1.7) gives rise to the following multidimensional analog of (2.2.9):

$$\phi_i(S_{w,i}^{n+1} - S_{w,i}^n) + \sum_{l \in \mathrm{adj}(i)} \lambda_{il} F_{il}(S_{w,i}^{n+1}, S_{w,l}^{n+1}) = 0. \tag{2.2.17}$$

Here, F_{il} is the flux (or velocity) from cell i to cell l, and $\lambda_{il} = \Delta t |\partial V_{il}|/|V_i|$, where $|\partial V_{il}|$ is the area of the surface separating cell i, and l, $|V_i|$ is the volume of cell i and Δt is the time step. For a conservative scheme we must have

$$F_{il}(u_i, u_l) = -F_{li}(u_l, u_i), \tag{2.2.18}$$

and for monotonicity we require that F_{il} be non-decreasing with respect to the first argument and non-increasing with respect to the second. This requirement is satisfied for two-phase flow problems, since we can reproduce the derivation in section 2.2.1

to obtain the flux function

$$F_{w,il} = \frac{\lambda_{w,il}}{\lambda_{T,il}} \left[q_{il} + K_{il}\lambda_{o,il}(g_w - g_o) \right]$$

and the upstream condition

$$\lambda_{p,il} = \begin{cases} \lambda_p(S_i) & \text{if } q_{il} + K_{il}(g_p - g_q)\lambda_{q,il} \geq 0, \\ \lambda_p(S_l) & \text{otherwise,} \end{cases}$$

for $p = o, w$, where $q_{il} = u_T^* \cdot \nu_{il}$ and $g_p = \gamma_p \nabla z \cdot \nu_{il}$. We show that a unique solution to (2.2.17) exists for any Δt if the following conditions hold:

1. The number of cells (control volumes) adjacent to cell i, $|\mathrm{adj}(i)|$, is bounded for all i;

2. The ratio $|\partial V_{il}|/|V_i|$ is bounded for all pairs of adjacent cells (i, l);

3. The quantity $\phi_i |V_i|$ is uniformly bounded away from zero for all i;

4. For any cell i, the total number of cells reachable from i in k steps is $O(k^p)$ for some fixed $p > 0$ (i.e. grows at most polynomially in k).

5. F_{il} is equicontinuous with the same Lipschitz constant for all pairs of adjacent cells (i, l).

Assumptions 1–4 are easily satisfied by regular Cartesian grids, and also by most unstructured grids of practical interest. From (2.2.18) we see that assumption 5 is satisfied as long as K_{il} is uniformly bounded over the domain, which is generally true for problems of practical interest. We justify these assumptions in section 2.3.7.

2.3 Existence and uniqueness of solutions for the discretized problems

In both model problems, we must solve a system of nonlinear equations ((2.2.9) and (2.2.17) respectively) for the unknowns $\{u_i^{n+1}\}$. In this section, we show that the classical nonlinear Jacobi and Gauss-Seidel processes both converge to a unique bounded solution, which provides an alternate constructive proof of the well-definedness of implicit monotone schemes. In addition, we show that Jacobi and Gauss-Seidel both converge for any starting point that is bounded by the initial data, which leads to a practical algorithm for computing the solution. In the interest of clarity, we first consider the following one-dimensional problem:

$$\phi_i(u_i^{n+1} - u_i^n) + \lambda(F_{i+1/2}^{n+1} - F_{i-1/2}^{n+1}) = 0, \qquad \lambda = \Delta t/\Delta x, \ \ i \in \mathbb{Z}. \tag{2.3.1}$$

We then extend the analysis to problems with spatially-varying coefficients, as well as problems in multiple dimensions.

2.3.1 Implicit monotone schemes

Consider a numerical scheme of the form (2.3.1), where $F_{i+1/2}$ denotes the numerical flux across the interface between cells i and $i+1$. This scheme approximates the 1D nonlinear conservation law

$$\phi(x)u_t + f(x,u)_x = 0, \qquad (x,t) \in \mathbb{R} \times \mathbb{R}^+, \tag{2.3.2}$$

which generalizes problem (2.1.9), (2.1.10) to the variable porosity and permeability case. For simplicity, we assume a three-point scheme

$$F_{i+1/2}^{n+1} = F_{i+1/2}(u_i^{n+1}, u_{i+1}^{n+1});$$

thus, the implicit stencil at cell i involves the value at cell i at time t^n, as well as the values at cells $i-1$, i and $i+1$ at the *future* time t^{n+1}. Given we are interested in handling flux functions of the type shown in Figure 2.1(b), we do *not* assume that the flux function $f(x,u)$ is monotonic in u, so that sonic points may be present. Assume that f and F are both locally Lipschitz continuous (but not necessarily differentiable), and that the numerical flux function $F_{i+1/2}$ is *consistent* with f in the sense that

$$F_{i+1/2}(u,u) = f(x_{i+1/2}, u). \tag{2.3.3}$$

For the purpose of this thesis, a 1D implicit scheme is said to an *implicit monotone scheme* if the following assumption is satisfied.

Assumption 1 (Monotonic fluxes)**.** For all $i \in \mathbb{Z}$, the numerical flux function $F_{i+1/2}$ is non-decreasing in the first argument and non-increasing in the second argument, i.e. for any w, we have $F_{i+1/2}(u,w) \le F_{i+1/2}(v,w)$ and $F_{i+1/2}(w,u) \ge F_{i+1/2}(w,v)$ whenever $u \le v$.

As shown in section 2.2.1, the fully implicit 1D problem satisfies this assumption. We show that residual functions corresponding to implicit monotone schemes are in fact M-functions in the sense of Rheinboldt [64]. This allows us to prove the existence and uniqueness of solutions via a convergent iterative process.

Remark. Assumption 1 also guarantees that the resulting residual function is an m-accretive operator in $\ell^1(\mathbb{Z})$ (see [33] for a proof). In general, m-accretive functions and M-functions are not equivalent concepts. Consider the space $X = L^1(\mathbb{R}^n)$, i.e., the (finite) n-dimensional vector space with the L^1-norm. Then A is an m-accretive operator if A is continuous and for any $u, v \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} (A(u)_i - A(v)_i)\,\mathrm{sgn}(u_i - v_i) \ge 0,$$

which is equivalent to diagonal dominance when A is linear (see Appendix B). On the other hand, M-functions are generalizations of M-matrices, i.e., A is a nonsingular M-matrix if (1) $a_{ii} > 0$, (2) $a_{ij} \le 0$ for $i \ne j$, and (3) A^{-1} has only non-negative entries. Thus, if

$$M_1 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \qquad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix},$$

then the function $f_1(x) = M_1 x$ is m-accretive but not an M-function, and the reverse is true for $f_2(x) = M_2 x$. We do not directly use m-accretivity in this work.

Remark. Assumption 1 implies that (2.3.1) is an E-scheme (cf. [60]), so it is at most first-order accurate.

2.3.2 Nonlinear Jacobi and Gauss-Seidel process

Suppose we want to solve a nonlinear system of algebraic equations $R(x) = 0$ for $x \in \mathbb{R}^N$, where $R = (r_1, \ldots, r_N)^T : \mathbb{R}^N \to \mathbb{R}^N$. Then we can consider the *nonlinear Gauss-Seidel process*:

$$\begin{aligned} &\text{Solve} \quad r_i(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^*, x_{i+1}^k, \ldots, x_N^k) = 0 \text{ for } x_i^*, \\ &\text{Set} \qquad x_i^{k+1} = x_i^*, \quad i = 1, \ldots, N, \quad k = 1, 2, \ldots, \end{aligned} \tag{2.3.4}$$

as well as the *nonlinear Jacobi process*:

$$\begin{aligned} &\text{Solve} \quad r_i(x_1^k, \ldots, x_{i-1}^k, x_i^*, x_{i+1}^k, \ldots, x_N^k) = 0 \text{ for } x_i^*, \\ &\text{Set} \qquad x_i^{k+1} = x_i^*, \quad i = 1, \ldots, N, \quad k = 1, 2, \ldots \end{aligned} \tag{2.3.5}$$

If R is continuous, then we know that whenever Jacobi or Gauss-Seidel converge, they have to converge to a solution x^* such that $R(x^*) = 0$. We would like to use the tools in [64] to show that (2.3.1) has a unique solution for any mesh ratio λ. However, since (2.3.1) is defined all $i \in \mathbb{Z}$, we need to extend Rheinboldt's results to include an appropriate class of infinite-dimensional systems in which the residual functions satisfy the following assumptions.

Assumption 2 (Preservation of bounded sets)**.** $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$ is a mapping between bounded sequences for which there exists an increasing function $\zeta : [0, \infty) \to$

$[0, \infty)$ such that

$$\|x\|_\infty \leq B \implies \|R(x)\| \leq \zeta(B).$$

Assumption 3 (Finite number of dependencies)**.** For each i, the residual function $r_i(x_1, x_2, \ldots)$ is non-constant with respect to a finite number (which can vary with i) of x_j.

In other words, the residual functions must come from a compact stencil and must preserve boundedness. With these assumptions, the nonlinear Gauss-Seidel process becomes

$$\begin{array}{ll} \text{Solve} & r_i(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i^*, x_{i+1}^k, x_{i+2}^k, \ldots) = 0 \text{ for } x_i^*, \\ \text{Set} & x_i^{k+1} = x_i^*, \quad i = 1, 2, \ldots, \quad k = 1, 2, \ldots, \end{array} \tag{2.3.6}$$

and the nonlinear Jacobi process becomes

$$\begin{array}{ll} \text{Solve} & r_i(x_1^k, \ldots, x_{i-1}^k, x_i^*, x_{i+1}^k, x_{i+2}^k, \ldots) = 0 \text{ for } x_i^*, \\ \text{Set} & x_i^{k+1} = x_i^*, \quad i = 1, 2, \ldots, \quad k = 1, 2, \ldots. \end{array} \tag{2.3.7}$$

The only differences between the above processes and (2.3.4)–(2.3.5) are that each Gauss-Seidel/Jacobi "sweep" now involves infinitely many variables and equations. These processes are well-defined because each r_i is assumed to depend on only finitely many arguments, so that for any given $i \in \mathbb{Z}, k \in \mathbb{N}$, the value of x_i^{k+1} can be obtained from a finite number of univariate solves. The main purpose of these assumptions is to ensure the residual function of the discretized PDE is an M-function. This would then allow us to prove the convergence of Jacobi and Gauss-Seidel iterations to a unique bounded solution.

2.3.3 M-function theory

M-functions are essentially generalizations of M-matrices in linear algebra. In the linear setting, it is well known (cf. [68]) that the Gauss-Seidel method applied to

$Ax = b$ converges for any right hand side b and starting point x_0 if A is an M-matrix. M-functions have similar properties with respect to the nonlinear Gauss-Seidel process, which is the subject of investigation in [64]. Here we provide extensions to the relevant definitions and theorems in [64] that would allow us to prove the existence and uniqueness of bounded solutions to (2.3.1).

For the remainder of the section, the natural partial ordering on $\ell^\infty(\mathbb{N})$ is written as $x \leq y$, i.e.,

$$x \leq y \qquad \Longleftrightarrow \qquad x_i \leq y_i, \quad \forall i \in \mathbb{N}.$$

We denote by e^i the unit basis vectors with the i-th component one and all others zero. The following definitions are essentially identical to those in [64], except the domain of definition has been changed from $\mathbb{R}^n$ to $\ell^\infty(\mathbb{N})$ to handle vectors of infinite length.

Definition 2.1. Let $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$.

1. R is *isotone* (or *antitone*) if, for all $x, y \in \ell^\infty(\mathbb{N})$, $x \leq y$ implies $R(x) \leq R(y)$ (or $R(x) \geq R(y)$). It is *strictly isotone* (or antitone) if $x < y$ implies $R(x) < R(y)$ (or $R(x) > R(y)$).

2. R is *inverse isotone* if, for all $x, y \in \ell^\infty(\mathbb{N})$, $R(x) \leq R(y)$ implies $x \leq y$.

3. R is (strictly) *diagonally isotone* if, for all $x \in \ell^\infty(\mathbb{N})$, the functions

$$\rho_{ii} : \mathbb{R} \to \mathbb{R}, \quad \rho_{ii}(t) = r_i(x + te^i), \quad i = 1, 2, \ldots \tag{2.3.8}$$

are (strictly) isotone.

4. R is *off-diagonally antitone* if, for any $x \in \ell^\infty(\mathbb{N})$, the functions

$$\rho_{ij} : \mathbb{R} \to \mathbb{R}, \quad \rho_{ij}(t) = r_i(x + te^j), \quad i \neq j, \quad i, j = 1, 2, \ldots \tag{2.3.9}$$

are antitone.

5. R is an *M-function* if R is inverse isotone and off-diagonally antitone.

One characterization of M-functions is given by Theorem 2.2, which generalizes the following result from matrix analysis: a square matrix A is an M-matrix if it has positive diagonal, non-positive off-diagonal, and is column diagonally dominant.

Theorem 2.2. *Suppose $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$ is off-diagonally antitone and satisfies Assumption 2 and 3. Suppose, for each $B > 0$, there exists a positive sequence $\{w_i^B\}$ such that*

1. $\sum_{i=1}^\infty w_i^B < \infty$,
2. *for any $\|x\|_\infty < B$, the function $Q(t) = (q_1(t), q_2(t), \dots)$ defined by*

$$q_i(t) = \sum_{j=1}^\infty w_j^B r_j(x + te^i)$$

is strictly isotone over the interval $t \in (t_{min}, t_{max})$, where

$$t_{min} = -B - \inf_i x_i, \quad t_{max} = B - \sup_i x_i.$$

Then R is an M-function.

Proof. The proof is an adaptation of the proof of Theorem 5.1 in [64], suitably modified to handle the infinite-dimensional case. Suppose $R(x) \leq R(y)$ for some $x, y \in \ell^\infty(\mathbb{N})$. Define the sets

$$N^- = \{i \in \mathbb{N} \mid y_i < x_i\}; \quad N^+ = \{i \in \mathbb{N} \mid y_i \geq x_i\}.$$

Suppose N^- is non-empty. For each $i \in N^-$, let $\gamma_i = (x_i - y_i)e^i$. We consider two cases:

1. If $|N^-| < \infty$, let $i_1 < i_2 < \dots < i_m$ be the elements of N^-, and define

$$z^0 = y, \quad z^1 = y + \gamma_{i_1}, \quad \dots, \quad z^m = y + \gamma_{i_1} + \dots + \gamma_{i_m},$$

and let $z^k = z^m = z$ for all $k > m$.

2. If $|N^-| = \infty$, let $i_1 < i_2 < \cdots$ be the elements of N^-, and define

$$z^0 = y, \quad z^1 = y + \gamma_{i_1}, \quad \ldots, \quad z^k = y + \gamma_{i_1} + \cdots + \gamma_{i_k}, \quad \ldots$$

and let $z = \{z_i\}$ be such that $z_i = \max\{x_i, y_i\}$.

Define $R^k := R(z^k)$ and $R^\infty = R(z)$. In either case, we have the following properties:

1. $\|z^k\|_\infty < B$ and $\|z\|_\infty < B$, where $B = \max\{\|x\|_\infty, \|y\|_\infty\}$. Hence, by Assumption 2, $\|R^k\|_\infty < \zeta(B)$ for all k (similarly for R^∞).
2. For each i, $z_i^k = z_i$ for large enough k, so by Assumption 3, $R_j^k \to R_j^\infty$ *pointwise* for each j.

Since $R_j^k < \zeta(B)$ for all j, k, each R^k is dominated by the constant sequence $G = (\zeta(B), \zeta(B), \ldots)$. Moreover $\sum_{j=1}^\infty w_j^B G_j < \infty$, so by the dominated convergence theorem (cf. [65]), we have

$$\sum_{j=1}^\infty w_j^B R_j^k \to \sum_{j=1}^\infty w_j^B R_j^\infty \quad \text{as } k \to \infty.$$

By the strict isotonicity of Q, we have

$$\sum_{j=1}^\infty w_j^B R_j^0 \le \sum_{j=1}^\infty w_j^B R_j^1 \le \cdots$$

with at least one strict inequality (since N^- is non-empty). Thus, we must have

$$\sum_{j=1}^\infty w_j^B r_j(y) = \sum_{j=1}^\infty w_j^B R_j^0 < \sum_{j=1}^\infty w_j^B R_j^\infty = \sum_{j=1}^\infty w_j^B r_j(z). \tag{2.3.10}$$

Now split the last sum into two parts:

$$\sum_{j=1}^\infty w_j^B r_j(z) = \sum_{j \in N^-} w_j^B r_j(z) + \sum_{j \in N^+} w_j^B r_j(z), \tag{2.3.11}$$

where the summation over N^+ may be empty. Then by off-diagonal antitonicity of R

(and invoking the dominated convergence theorem whenever necessary), we can show similarly that

$$\sum_{j\in N^-} w_j^B r_j(z) \le \sum_{j\in N^-} w_j^B r_j(x), \qquad \sum_{j\in N^+} w_j^B r_j(z) \le \sum_{j\in N^+} w_j^B r_j(y), \tag{2.3.12}$$

using the fact that $z-x$ and $z-y$ vanish on N^- and N^+ respectively. Combining equations (2.3.10)–(2.3.12) gives

$$\sum_{j=1}^{\infty} w_j^B r_j(y) < \sum_{j\in N^-} w_j^B r_j(x) + \sum_{j\in N^+} w_j^B r_j(y), \tag{2.3.13}$$

which implies

$$\sum_{j\in N^-} w_j^B r_j(y) < \sum_{j\in N^-} w_j^B r_j(x). \tag{2.3.14}$$

Thus, we must have $r_j(y) < r_j(x)$ for some $j \in N^-$, which contradicts the hypothesis $R(x) \le R(y)$. Hence N^- must be empty, so $x \le y$. □

The above theorem, together with the definition of M-functions, immediately imply the following corollary.

Corollary 2.3. *Let R satisfy the hypotheses of Theorem 2.2. Let $z \in \ell^\infty(\mathbb{N})$. Then there is at most one bounded solution to the equation $R(x) = z$.*

Remark. In the context of discretized PDEs one normally assumes tacitly that the solution of interest must be bounded; this can be regarded as a boundary condition "at infinity". However, since such boundary conditions are not explicitly stated in the definition of M-functions, one must be careful to exclude any parasitic unbounded solutions that may arise. In fact, the solution is not necessarily unique if we allow unbounded solutions. Consider the linear function $R = (r_1, r_2, \dots)$ defined by $r_i(x) = x_i - \alpha x_{i+1}$ for $|\alpha| < 1$. Then for any $\|x\|_\infty < \infty$, we have $\|R(x)\|_\infty \le (1+\alpha)\|x\|_\infty$, so that Assumption 2 is satisfied. Assumption 3 (finitely many dependencies) is also satisfied because each r_i is only non-constant with respect to two components of x.

Finally, if we let $w_j^B = \beta^j$ for any $|\alpha| < \beta < 1$, then $\sum_j \beta^j < \infty$ and

$$\begin{aligned} q_i(t) &= \sum_{j=1}^{\infty} \beta^j r_j(x + te^i) \\ &= \sum_{j=1}^{\infty} \beta^j \left[x_j + t\delta_{ij} - \alpha(x_{j+1} + t\delta_{i,j+1}) \right] \\ &= (\beta - \alpha)\beta^{i-1}t + \beta x_1 + (\beta - \alpha) \sum_{j=2}^{\infty} \beta^{j-1} x_j, \end{aligned}$$

so $q_i(t)$ is well-defined and is strictly increasing with respect to t whenever $\|x\|_\infty < \infty$. So the hypotheses of Theorem 2.2 are satisfied, and hence $x = 0$ is the only bounded solution of $R(x) = 0$. However, unbounded solutions of the form $y = \{K\alpha^{-i}\}$, $K \neq 0$ also satisfy $R(y) = 0$, so the theorem does not preclude these possibilities.

2.3.4 Convergence of nonlinear Jacobi and Gauss-Seidel

It turns out that the hypotheses of Theorem 2.2 are enough to ensure convergence of nonlinear Jacobi and Gauss-Seidel for certain starting points described below. The following result is essentially Theorem 3.1 in [64], with modified hypotheses to accommodate ℓ^∞-bounded vectors with infinitely many components. The proof in [64] goes through verbatim, but is reproduced here for completeness. Note that by Assumption 3, each r_i depends on only finitely many arguments, so the standard arguments on limits, continuity and antitonicity hold without additional complications when they are used on individual components of R.

Theorem 2.4 (Rheinboldt). *Let $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$ satisfy the hypotheses of Theorem 2.2. Suppose for some $z \in \ell^\infty(\mathbb{N})$ there exist points $x^0, y^0 \in \ell^\infty(\mathbb{N})$ such that*

$$x^0 \leq y^0, \qquad R(x^0) \leq z \leq R(y^0).$$

Then the nonlinear Gauss-Seidel and Jacobi iterates $\{y^k\}$ and $\{x^k\}$, given by (2.3.6) *and* (2.3.7) *and starting from* y^0 *and* x^0*, respectively, are uniquely defined and satisfy*

$$x^0 \leq x^k \leq x^{k+1} \leq y^{k+1} \leq y^k \leq y^0, \qquad R(x^k) \leq z \leq R(y^k) \tag{2.3.15}$$

for all $k \geq 0$*. In addition, the pointwise limits*

$$\lim_{k\to\infty} x^k = \lim_{k\to\infty} y^k = x^* \tag{2.3.16}$$

exist, and $R(x^*) = z$*.*

First we need the following lemma (which is part of Theorem 2.10 in [64]).

Lemma 2.5. *Let* $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$ *be an* M*-function. Then* R *is strictly diagonally isotone.*

Proof. Suppose that for some $x \in \ell^\infty(\mathbb{N})$, $s, t \in \mathbb{R}$, $s > t$ and index i we have $r_i(x + se^i) \leq r_i(x + te^i)$. The off-diagonal antitonicity then implies that

$$r_j(x + se^i) \leq r_j(x + te^i), \qquad j \neq i,$$

or, altogether, that $R(x + se^i) \leq R(x + te^i)$. By inverse isotonicity this leads to the contradiction $s \leq t$, which shows that R must be strictly diagonally isotone. □

Proof of Theorem 2.4. We present only the proof for convergence of Gauss-Seidel; the proof for Jacobi is similar. We proceed by induction and suppose that for some $k \geq 0$ and $i \geq 1$,

$$x^0 \leq x^k \leq y^k \leq y^0, \qquad R(x^k) \leq z \leq R(y^k), \tag{2.3.17a}$$

$$x_j^k \leq x_j^{k+1} \leq y_j^{k+1} \leq y_j^k, \qquad j = 1, \ldots, i-1, \tag{2.3.17b}$$

where for $i = 1$ the relation (2.3.17b) is vacuous. Clearly, (2.3.17) is valid for $k = 0$ and $i = 1$. Define the functions

$$\alpha(s) = r_i(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, s, x_{i+1}^k, x_{i+2}^k, \ldots)$$
$$\beta(s) = r_i(y_1^{k+1}, \ldots, y_{i-1}^{k+1}, s, y_{i+1}^k, x_{i+2}^k, \ldots)$$

for $s \in [x_i^0, y_i^0]$. From (2.3.17) and the off-diagonal antitonicity of R, we then find that

$$\beta(s) \leq \alpha(s), \qquad s \in [x_i^0, y_i^0], \tag{2.3.18}$$

and

$$\beta(x_i^k) \leq \alpha(x_i^k) \leq r_i(x^k) \leq z_i \leq r_i(y^k) \leq \beta(y_i^k) \leq \alpha(y_i^k). \tag{2.3.19}$$

By the continuity and strict isotonicity of α and β (since R is an M-function and hence strictly diagonally isotone), (2.3.19) implies the existence of unique $\hat{y}_i^k$ and $\hat{x}_i^k$ for which

$$\beta(\hat{y}_i^k) = z_i = \alpha(\hat{x}_i^k), \qquad x_i^k \leq \hat{x}_i^k \leq \hat{y}_i^k \leq y_i^k,$$

where the relation $\hat{x}_i^k \leq \hat{y}_i^k$ is a consequence of (2.3.18). But $x_i^{k+1} = \hat{x}_i^k$ and $y_i^{k+1} = \hat{y}_i^k$ by definition, so we have proved (2.3.17b) for $j = 1, \ldots, i$. By induction (2.3.17b) holds for all $i \in \mathbb{N}$, and hence

$$x^k \leq x^{k+1} \leq y^{k+1} \leq y^k.$$

From this it follows again from off-diagonal antitonicity that

$$r_i(y^{k+1}) \geq r_i(y_1^{k+1}, \ldots, y_i^{k+1}, y_{i+1}^k, y_{i+2}^k \ldots) = z_i$$

and similarly that

$$r_i(x^{k+1}) \leq r_i(x_1^{k+1}, \ldots, x_i^{k+1}, x_{i+1}^k, x_{i+2}^k \ldots) = z_i.$$

This completes the induction on k and hence the proof of (2.3.15). Applying the monotone convergence theorem for sequences, we conclude that the *pointwise* limits

$$\lim_{k\to\infty} x_j^k = x_j^* \leq y_j^* = \lim_{k\to\infty} y_j^k$$

exist for each j, which allows us to define $x^* = \{x_j^*\}$ and $y^* = \{y_j^*\}$. Since each r_i is continuous and depends on only finitely many arguments, the definition of the

Gauss-Seidel process then implies $r_i(x^*) = r_i(y^*) = z_i$ for each i, and hence

$$R(x^*) = R(y^*) = z.$$

Since both x^* and y^* are bounded, Corollary 2.3 implies that they are equal, completing the proof. □

2.3.5 Well-definedness of implicit monotone schemes

Using the theory in the last two sections, we can now prove that implicit monotone schemes (i.e., implicit schemes whose flux functions satisfy Assumption 1) are well-defined for bounded initial conditions. What we need to show is that the residual functions satisfy the hypotheses of Theorem 2.4. In the interest of clarity, in this section we only show convergence of the iterative schemes for problems whose coefficients do not vary in space (i.e., corresponding to the conservation law $u_t + f(u)_x = 0$, discretized on a uniform spatial grid). In the next section, we state the additional assumptions on ϕ_i and $F_{i+1/2}$ that are required for the spatially-varying case.

Theorem 2.6. *Consider the numerical scheme* (2.3.1) *with the numerical flux given by*

$$F_{i+1/2}^{n+1} = F(u_i^{n+1}, u_{i+1}^{n+1}),$$

where $F : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is locally Lipschitz continuous and satisfies Assumption 1, i.e., non-decreasing in the first argument and non-increasing in the second. Assume that the initial condition $\{u_i^0\}_{i=-\infty}^{\infty}$ is bounded. Then (2.3.1) *has a unique bounded solution $\{u_i^{n+1}\}$ for $n = 0, 1, 2, \ldots$. Moreover, this bounded solution satisfies the estimate*

$$\inf_{j\in\mathbb{Z}} u_j^n \le u_i^{n+1} \le \sup_{j\in\mathbb{Z}} u_j^n \tag{2.3.20}$$

for all $i \in \mathbb{Z}$.

Proof. The strategy is to start by defining an ordering for the Gauss-Seidel sweeps, i.e., by permuting the equations and variables so that the spatial indices go from 1 to ∞ rather than from $-\infty$ to ∞. After that, it suffices to check that all the hypotheses of Theorem 2.4 are satisfied for this ordering.

1. For $j = 1, 2, \ldots$, define $\sigma(j) = (-1)^j \lfloor j/2 \rfloor$, i.e. σ maps $\{1, 2, 3, 4, 5, \ldots\}$ to $\{0, 1, -1, 2, -2, \ldots\}$. Let τ be the inverse map, such that $\tau(\sigma(j)) = j$. Define $R : \ell^\infty(\mathbb{N}) \to \ell^\infty(\mathbb{N})$ to be the reordered (and rescaled) set of residual equations, i.e.,
$$r_j(v) = \frac{v_j - u^n_{\sigma(j)}}{\lambda} + F(v_j, v_{\tau(\sigma(j)+1)}) - F(v_{\tau(\sigma(j)-1)}, v_j), \tag{2.3.21}$$
where $v_j = u^{n+1}_{\sigma(j)}$.

2. Since F is locally Lipschitz continuous, it is Lipschitz continuous over any compact set, so for any $B > 0$ there exists K_B (which can be chosen to be increasing with B) such that for any $(x, y) \in [-B, B] \times [-B, B]$,
$$|F(x, y) - F(0, 0)| \leq K_B(|x| + |y|) \leq 2K_B \cdot B.$$
Thus, for any $\|v\|_\infty \leq B$, we have $|r_j(v)| \leq \zeta(B)$ for all j, where
$$\zeta(B) = (2/\lambda + 4K_B)B.$$
Hence Assumption 2 is satisfied. Moreover, since each r_j depends only on v_j, $v_{\tau(\sigma(j)-1)}$ and $v_{\tau(\sigma(j)+1)}$, Assumption 3 (finite number of dependencies) is also satisfied.

3. By Assumption 1 (monotonic fluxes), F is clearly off-diagonally antitone. To satisfy the remaining hypotheses of Theorem 2.2, let $\{w^B_j\}$ take the form $w^B_j = \beta^{|\sigma(j)|}$ for some $0 < \beta < 1$, so that $\sum_{j=1}^\infty w^B_j < \infty$. An easy calculation shows that
$$q_i(t) := \sum_{j=1}^\infty w^B_j r_j(v + te^i) = \tilde{q}_i(t) + \sum_{j=1}^\infty w^B_j r_j(v),$$
where
$$\begin{aligned}\tilde{q}_i(t) = w^B_i t/\lambda &+ (w^B_i - w^B_{\tau(\sigma(i)+1)})\big[F(v_i + t, v_{\tau(\sigma(i)+1)}) - F(v_i, v_{\tau(\sigma(i)+1)})\big] \\ &+ (w^B_{\tau(\sigma(i)-1)} - w^B_i)\big[F(v_{\tau(\sigma(i)-1)}, v_i + t) - F(v_{\tau(\sigma(i)-1)}, v_i)\big].\end{aligned}$$

By the definition of w_i^B, we see that

$$|w_i^B - w_{\tau(\sigma(i)\pm 1)}^B| \leq \beta^{|\sigma(i)|-1}(1-\beta),$$

which, when combined with the local Lipschitz continuity of F, gives

$$\beta^{|\sigma(i)|-1}\big[\beta t/\lambda - 2(1-\beta)K_B|t|\big] \leq \tilde{q}_i(t) \leq \beta^{|\sigma(i)|-1}\big[\beta t/\lambda + 2(1-\beta)K_B|t|\big].$$

Hence, $\tilde{q}_i(t)$ is strictly isotone whenever

$$\beta/\lambda > 2(1-\beta)K_B,$$

so picking

$$\frac{2\lambda K_B}{1+2\lambda K_B} < \beta < 1 \tag{2.3.22}$$

ensures isotonicity for $\tilde{q}_i(t)$ (and hence $q_i(t)$) for all i, as required in Theorem 2.2. (Note that the choice of β depends on B.)

4. We need to choose starting points x^0 and y^0 that satisfy the requirements of Theorem 2.4. Let x^0 and y^0 both be constant sequences with

$$x_i^0 = \inf_{j\in\mathbb{Z}} u_j^n, \qquad y_i^0 = \sup_{j\in\mathbb{Z}} u_j, \qquad \forall i \in \mathbb{N}.$$

Then clearly $x^0 \leq y^0$, and for all $i \in \mathbb{N}$,

$$r_i(x^0) = x_i^0 - u_{\sigma(i)}^n = \inf_{j\in\mathbb{Z}} u_j^n - u_{\sigma(i)}^n \leq 0,$$
$$r_i(y^0) = y_i^0 - u_{\sigma(i)}^n = \sup_{j\in\mathbb{Z}} u_j^n - u_{\sigma(i)}^n \geq 0,$$

so $R(x^0) \leq 0 \leq R(y^0)$. Thus, by Theorem 2.4, the nonlinear Gauss-Seidel iterates $\{y^k\}$ and $\{x^k\}$ both converge (pointwise) to the unique solution x^* with $R(x^*) = 0$; hence, a unique solution to (2.3.1) exists, i.e.,

$$u_i^{n+1} = x_{\tau(i)}^*.$$

Moreover, we know that $x^0 \leq x^* \leq y^0$, which immediately implies (2.3.20). □

Remarks.

1. Note that the initial condition $\{u_i^0\}_{i=-\infty}^{\infty}$ is not assumed to be in ℓ^1 nor in BV, so this result is somewhat more general than classical results that use Crandall-Liggett theory.

2. Note that the definition of an M-function is invariant under symmetric permutations, i.e., $R(x)$ is an M-function if and only if $\sigma R(\sigma x)$ is also an M-function for any permutation $\sigma : \mathbb{N} \to \mathbb{N}$. Thus, the Gauss-Seidel process will converge regardless of the way the ordering is chosen in step 1 of the proof. However, we show in the next section that the *rate* of convergence is sensitive to the ordering.

In fact, one can show that the nonlinear Jacobi and Gauss-Seidel processes converge for any starting point $\{z_i^{(0)}\}$ that is bounded by the initial data $\{u_i^n\}$. (In the sequel, superscripts in brackets indicate iterates within the Gauss-Seidel process, and superscripts without brackets indicate the time level in the numerical scheme.)

Theorem 2.7. *Assume the hypotheses of Theorem 2.6. Suppose the initial guess $\{z_i^{(0)}\}$ satisfies*

$$\inf_{j\in\mathbb{Z}} u_j^n \leq z_i^{(0)} \leq \sup_{j\in\mathbb{Z}} u_j^n \tag{2.3.23}$$

for all $i \in \mathbb{Z}$. Then the nonlinear Jacobi and Gauss-Seidel processes (2.3.6) *and* (2.3.7) *are well-defined and converge to the unique bounded solution of* (2.3.1).

Proof. Again we only show convergence for the Gauss-Seidel process, since the proof for Jacobi is similar. Denote $\underline{u} = \inf_{j\in\mathbb{Z}} u_j^n$ and $\overline{u} = \sup_{j\in\mathbb{Z}} u_j^n$. First, we show that the Gauss-Seidel iterates are well-defined and that $\underline{u} \leq u_j^{(k)} \leq \overline{u}$ for all j, k. At each step we need to solve

$$r_j(z_j^*) = z_j^* - u_j^n + \lambda \left[F(z_j^*, z_{j+1}) - F(z_{j-1}, z^*j)\right] = 0, \tag{2.3.24}$$

where $z_{j\pm1} = z_{j\pm1}^{(k)}$ or $z_{j\pm1}^{(k+1)}$ depending on the ordering of the Gauss-Seidel sweep, which by induction must lie between $\underline{u}$ and $\overline{u}$. But

$$\begin{aligned} r_j(\underline{u}) &= \underline{u} - u_j^n + \lambda \left[F(\underline{u}, z_{j+1}) - F(z_{j-1}, \underline{u}) \right] \\ &\leq 0 + \lambda \left[F(\underline{u}, \underline{u}) - F(\underline{u}, \underline{u}) \right] \leq 0, \end{aligned}$$

where the last inequality follows from Assumption 1. Similarly one obtains $r_j(\overline{u}) \geq 0$, so by continuity of F (and hence r_j) there must exist a solution z_j^* to (2.3.24), which by Lemma 2.5 must be unique. Hence, by induction, the Gauss-Seidel iterates are well-defined and are bounded above and below by $\overline{u}$ and $\underline{u}$ respectively.

Now consider the Gauss-Seidel iterates $\{x_j^{(k)}\}$ and $\{y_j^{(k)}\}$ with initial guess $x_j^{(0)} = \underline{u}$ and $y_j^{(0)} = \overline{u}$ for all j. By Theorem 2.6 these iterates converge pointwise to the same solution $\{x_j^*\}$. We show inductively that $x^{(k)} \leq z^{(k)} \leq y^{(k)}$ for all k, which would imply that $z_j^{(k)} \to x_j^*$ pointwise. Using the same reordering as in Theorem 2.6, assume that for some $k \geq 0$ and $i \geq 1$ we have

$$y^{(k)} \geq z^{(k)} \geq x^{(k)}, \qquad y_j^{(k+1)} \geq z_j^{(k+1)} \geq x_j^{(k+1)}, \qquad j = 1, \ldots, i-1,$$

which is valid for $k = 0$ and $i = 1$. Then by the same boundedness and antitonicity arguments as in Theorem 2.4, we have

$$\begin{aligned} r_i(y_1^{(k+1)}, \ldots, y_{i-1}^{(k+1)}, y_i^{(k+1)}, y_{i+1}^{(k)}, \ldots) = 0 &= r_i(z_1^{(k+1)}, \ldots, z_{i-1}^{(k+1)}, z_i^{(k+1)}, z_{i+1}^{(k)}, \ldots) \\ &\geq r_i(y_1^{(k+1)}, \ldots, y_{i-1}^{(k+1)}, z_i^{(k+1)}, y_{i+1}^{(k)}, \ldots), \end{aligned}$$

which, together with the strict diagonal isotonicity or r_i, implies that $y_i^{(k+1)} \geq z_i^{(k+1)}$. Similarly it follows that $z_i^{(k+1)} \leq x_i^{(k+1)}$. This completes the induction, and hence $z_j^{(k)} \to x_j^*$ pointwise. □

In other words, the nonlinear Gauss-Seidel process converges if we use $\{u_j^n\}$ (i.e., the solution from the previous time step) as an initial guess. For small to moderate timestep sizes, one generally expects the solutions between consecutive time steps to be close to each other, so in practice using $\{u_j^n\}$ results in much faster convergence than either $\underline{u}$ or $\overline{u}$ as an initial guess.

2.3.6 Rate of convergence of the nonlinear processes

So far we have proven that the nonlinear Gauss-Seidel and Jacobi processes both converge globally when applied to residual functions arising from implicit monotone schemes, but we have not investigated how fast these processes converge. For this purpose, let us reconsider the finite-dimensional case, i.e., when $R : \mathbb{R}^N \to R^N$ is given by

$$r_i(u^{n+1}) = \phi_i(u_i^{n+1} - u_i^n) + \lambda(F_{i+1/2}(u_i^{n+1}, u_{i+1}^{n+1}) - F_{i-1/2}(u_{i-1}^{n+1}, u_i^{n+1})) \qquad (2.3.25)$$

for $i = 1, \ldots, N$, and the finite versions of the Gauss-Seidel and Jacobi processes ((2.3.4) and (2.3.5)) are used. It is well known [59] that for a convergent fixed-point iteration $x^{n+1} = Gx^n$, the asymptotic rate of convergence is given by $\rho(G'(x^*))$, the spectral radius of the Jacobian matrix evaluated at the solution x^*. Moreover, superlinear convergence is obtained when $\rho(G'(x^*)) = 0$. The following lemma gives the rate of convergence for the nonlinear Gauss-Seidel and Jacobi processes.

Lemma 2.8. *Suppose the residual function $R : D \subset \mathbb{R}^N \to \mathbb{R}^N$ is an M-function that is continuously differentiable at x^*. Let the Jacobian matrix be written as $R'(x^*) = D - L - U$, where D is a diagonal matrix and L, U are strictly lower and upper triangular respectively. Then the asymptotic rates of convergence for the nonlinear Gauss-Seidel and Jacobi processes (*(2.3.4) *and* (2.3.5)*) are given by ρ_{GS} and ρ_J respectively, where*

$$\rho_{GS} = \rho((D - L)^{-1}U), \qquad \rho_J = \rho(D^{-1}(L + U)).$$

Proof. Let G denote the Gauss-Seidel operator, i.e., $y := x^{k+1} = Gx^k$, where x^{k+1} is defined implicitly as a function of x^k by (2.3.4). Then implicit differentiation gives, for each $j = 1, \dots, N$,

$$\begin{aligned}
&\frac{\partial r_1}{\partial y_1}\frac{\partial y_1}{\partial x_j} && +\frac{\partial r_1}{\partial x_j} = 0\\
&\frac{\partial r_2}{\partial y_1}\frac{\partial y_1}{\partial x_j} + \frac{\partial r_2}{\partial y_2}\frac{\partial y_2}{\partial x_j} && +\frac{\partial r_2}{\partial x_j} = 0\\
& && \vdots\\
&\frac{\partial r_N}{\partial y_1}\frac{\partial y_1}{\partial x_j} + \cdots + \frac{\partial r_N}{\partial y_N}\frac{\partial y_N}{\partial x_j} && +\frac{\partial r_N}{\partial x_j} = 0
\end{aligned}$$

We can rewrite the above in matrix form as

$$\big[D(y) - L(y)\big]\frac{\partial y}{\partial x} - U(x) = 0,$$

where D, L and U are the diagonal, strict lower-triangular and strict upper-triangular part of $\partial R/\partial x$ respectively. Since R is an M-function, $(D(y) + L(y))$ is nonsingular for all $y \in D$. Thus, G' is given by

$$G'(x) = \frac{\partial y}{\partial x} = \big[D(y(x)) - L(y(x))\big]^{-1}U(x).$$

Since R is continuously differentiable at x^*, letting $x, y \to x^*$ shows that the asymptotic rate of convergence is given by $\rho((D-L)^{-1}U)$, as required. The argument for the Jacobi process is similar. □

In other words, the rates of convergence of the nonlinear processes are exactly the same as the rates for the corresponding linear processes applied to the Jacobian matrix of the residual function. For the residual function (2.3.25), the Jacobian matrix has the following tridiagonal form:

$$\frac{\partial R}{\partial u} = \begin{bmatrix} d_1 & f_1 & & \\ e_2 & d_2 & \ddots & \\ & \ddots & \ddots & f_{N-1} \\ & & e_N & d_N \end{bmatrix},$$

where

$$\begin{aligned} d_i &= \phi_i + \lambda \left(\frac{\partial F_{i+1/2}}{\partial u_i} - \frac{\partial F_{i-1/2}}{\partial u_i} \right) > 0, \\ e_i &= -\lambda \frac{\partial F_{i-1/2}}{\partial u_{i-1}} \leq 0, \\ f_i &= \lambda \frac{\partial F_{i+1/2}}{\partial u_{i+1}} \leq 0. \end{aligned}$$

Thus, $d_i = \phi_i - e_{i+1} - f_{i-1}$, so that $\partial R / \partial u$ is a column diagonally dominant matrix. This guarantees that both ρ_{GS} and ρ_J are strictly less than 1. Since $\partial R/\partial u$ is generally not a diagonal matrix, it is clear that nonlinear Jacobi converges at most linearly. One can compute an upper bound for ρ_J as follows. We have

$$\begin{aligned} \rho_J &= \rho(D^{-1}(L+U)) = \rho((L+U)D^{-1}) \\ &\leq \left\| (L+U)D^{-1} \right\|_1 = \max_i \left(\frac{e_{i+1} + f_{i-1}}{d_i} \right) \\ &\leq 1 - \frac{\min \phi_i}{\max d_i}. \end{aligned}$$

Since $-\log \rho_J \approx \phi_{\min} / \max d_i$, it follows that $-\log \rho_J$ is roughly inversely proportional to the mesh ratio λ, especially when λ (and equivalently Δt) is large. Thus, one expects Jacobi to take roughly twice as many iterations to converge when one doubles the time-step size while fixing the spatial grid (or, equivalently, when the grid is refined by a factor of two while Δt is kept constant).

For Gauss-Seidel, we exploit the fact that $\partial R / \partial u$ is tridiagonal. For this class of matrices (and in fact, for any *consistently ordered matrices* in the sense of Young [85]), the following theorem holds [68].

Theorem 2.9. *Let A be a consistently ordered matrix such that $a_{ii} \neq 0$ for $i = 1, \ldots, N$, and let the SOR parameter ω be nonzero. Then, if λ is a nonzero eigenvalue of the SOR iteration matrix G_{SOR}, any scalar μ such that*

$$(\lambda + \omega - 1)^2 = \lambda \omega^2 \mu^2 \tag{2.3.26}$$

is an eigenvalue of the Jacobi iteration matrix G_J. Conversely, if μ is an eigenvalue of G_J and if a scalar λ satisfies (2.3.26)*, then λ is an eigenvalue of G_{SOR}.*

Since Gauss-Seidel is simply SOR with $\omega = 1$, it follows that either $\rho_{GS} = 0$ or $\rho_{GS} = \rho_J^2$. The former happens when $f_i = 0$ for $i = 1, \dots, N-1$, i.e., when $\partial R/\partial u$ is lower triangular. In this case, Gauss-Seidel converges in one iteration (i.e., superlinearly), since the nonlinear system is actually decoupled and Gauss-Seidel is essentially just a forward substitution. For 1D porous media flow, this occurs when flow is purely cocurrent and the numerical scheme reverts to single-point upstreaming. When countercurrent flow is present, there is no symmetric permutation that would render $\partial R/\partial u$ lower triangular, so Gauss-Seidel also converges linearly, requiring about half as many iterations as Jacobi.

2.3.7 Extensions

In this section we show how to extend the results of Theorems 2.6 and 2.7 to deal with:

1. conservation laws with non-uniform spatial grids,
2. spatially-varying flux functions,
3. problems in which the flux functions are only defined over a closed interval $I \subset \mathbb{R}$, and
4. problems in multiple dimensions.

Non-uniform grids and spatially-varying flux functions

Consider again the fully-implicit discretization (2.3.1):

$$\phi_i(u_i^{n+1} - u_i^n) + \lambda(F_{i+1/2}^{n+1} - F_{i-1/2}^{n+1}) = 0, \qquad \lambda = \Delta t/\Delta x, \;\; i \in \mathbb{Z},$$

with a spatially-varying ϕ_i and $F_{i+1/2}$. We assume that $0 < \phi_i \leq 1$. Notice that the non-uniform grid case is automatically included: for any non-uniform discretization of the form

$$\frac{\tilde{\phi}_i(u_i^{n+1} - u_i^n)}{\Delta t} + \frac{F_{i+1/2}^{n+1} - F_{i-1/2}^{n+1}}{\Delta x_i} = 0, \tag{2.3.27}$$

we can multiply (2.3.27) by $\Delta t \Delta x_i / \Delta x_{\max}$ to recover the form of (2.3.1) with

$$\phi_i = \tilde{\phi}_i \Delta x_i / \Delta x_{\max}, \qquad \lambda = \Delta t / \Delta x_{\max}.$$

To ensure convergence of the Jacobi and Gauss-Seidel processes, we need the following assumptions:

1. The family of flux functions $\{F_{i+1/2}\}_{i=-\infty}^{\infty}$ is equicontinuous [65] with the same Lipschitz constant K_B;
2. $\{\phi_i\}$ is uniformly bounded away from zero, i.e. there exists $\phi_{\min} > 0$ such that $\phi_i \geq \phi_{\min}$ for all $i \in \mathbb{Z}$.

While the equicontinuity condition may appear severe, it is usually satisfied in practice because the spatially-varying coefficients (e.g. $K(x)$ in (2.1.10)) tend to be uniformly bounded, ensuring equicontinuity in the flux functions. With the above assumptions, we can mimic Theorem 2.6 exactly by replacing λ with λ/ϕ_i. Then the proof goes through verbatim, except for (2.3.22), which must be modified to

$$\frac{2\lambda K_B}{\phi_{\min} + 2\lambda K_B} < \beta < 1. \tag{2.3.28}$$

Bounded admissible solutions

Formally, Theorem 2.6 requires the discrete flux function $F(u_i, u_{i+1})$ to be defined on $\mathbb{R} \times \mathbb{R}$. In practice one may want to solve problems for which the flux function f is only defined on an interval $[u_{\min}, u_{\max}]$ rather than on all of $\mathbb{R}$, so states outside these physical bounds are not admissible. For instance, in the two-phase flow problem, we must have $S_i \in [0, 1]$ for all i, and the flux function $f(S)$ in (2.1.10) is not even defined outside this range. Fortunately, the estimate (2.3.20) ensures that as long as the initial conditions are within physical bounds, so will the solution remain for subsequent time steps $n > 0$. Thus, in order to apply Theorem 2.6 to these problems, one can *formally extend* the domain of definition of the flux function f to $\mathbb{R}$ by defining, for instance,

$$\tilde{f}(u) = \begin{cases} f(u_{\min}), & u < u_{\min}, \\ f(u), & u_{\min} \le u \le u_{\max}, \\ f(u_{\max}), & u > u_{\max}, \end{cases}$$

and similarly for the discrete flux $F(u, v)$. Since all the Gauss-Seidel iterates $\{y^k\}$ and $\{x^k\}$ satisfy the bound $x^0 \le x^k \le y^k \le y^0$, the exact manner in which the extension is defined is unimportant as long as the monotonicity property (Assumption 1) is satisfied.

Multiple dimensions

The M-function analysis above can be extended to scalar conservation laws in multiple dimensions. Consider once again the conservative, implicit monotone scheme

$$\phi_i(u_i^{n+1} - u_i^n) + \sum_{l \in \text{adj}(i)} \lambda_{il} F_{il}(u_i^{n+1}, u_l^{n+1}) = 0, \quad i \in \mathcal{I}, \tag{2.3.29}$$

of which the SEQ problem is an example. Recall that F_{il} is the flux from cell i to cell l, $\lambda_{il} = \Delta t |\partial V_{il}| / |V_i|$, where $|\partial V_{il}|$ is the area of the surface separating cell i and l, $|V_i|$ is the volume of cell i and Δt is the time step. In order to mimic Theorem 2.6, we need the following assumption on the numerical flux:

1. F_{il} is equicontinuous with the same Lipschitz constant for all pairs of adjacent cells (i, l),

as well as these assumptions on the grid:

2. The number of cells (control volumes) adjacent to cell i, $|\text{adj}(i)|$, is bounded for all i;

3. The ratio $|\partial V_{il}| / |V_i|$ is bounded for all pairs of adjacent cells (i, l);

4. The quantity $\phi_i |V_i|$ is uniformly bounded away from zero for all i;

5. For any cell i, the total number of cells reachable from i in k steps is $O(k^p)$ for some fixed $p > 0$ (i.e. grows at most polynomially in k).

Items (1) and (4) are analogous to the conditions stated in the non-uniform grid case, whereas the other conditions are new. These assumptions ensure that the residual functions are all bounded and have the same Lipschitz constant over the set $\{u \in \ell^\infty(N) \,|\, \|u\|_\infty < B\}$. The polynomial growth assumption (5) allows us to assign the weights $\{w_i^B\}$ to each cell i in the following manner: pick any node i_0 and let $w_i^B = \beta^{d(i_0,i)}$, where $d(i,j)$ is the shortest distance between node i and j in the graph-theoretic sense. Since the number of cells within k steps of i_0 grows polynomially in k, the series $\sum_i w_i^B$ converges for any $0 < \beta < 1$, so β can be chosen the same way as in step 3 of Theorem 2.6 and the same argument will hold.

2.3.8 Maximum principle

We conclude this section by proving a stronger version of (2.3.20) that is satisfied by implicit monotone schemes, as well as any Gauss-Seidel iterates.

Proposition 2.10 (Maximum principle)**.** *Suppose u^* solves the problem*

$$u^* - u^0 + \lambda\big[F(u^*, u_+) - F(u_-, u^*)\big] = 0,$$

where F satisfies Assumption 1. Then u^* satisfies

$$\min\{u^0, u_-, u_+\} \le u^* \le \max\{u^0, u_-, u_+\}. \tag{2.3.30}$$

Proof. If u^* is equal to any one of u^0, u_-, u_+, there is nothing to prove. So assume $u^* \notin \{u^0, u_-, u_+\}$. Define

$$C = \frac{F(u^*, u^*) - F(u^*, u_+)}{u_+ - u^*} \ge 0 \tag{2.3.31}$$

$$D = \frac{F(u^*, u^*) - F(u_-, u^*)}{u^* - u_-} \ge 0. \tag{2.3.32}$$

The non-negativity of C and D follows from Assumption 1. Then

$$\begin{aligned}0 &= u^* - u^0 + \lambda\big[F(u^*, u_+) - F(u_-, u^*)\big]\\ &= u^* - u^0 + \lambda\big[F(u^*, u_+) - F(u^*, u^*) + F(u^*, u^*) - F(u_-, u^*)\big]\\ &= (u^* - u^0) + \lambda C(u^* - u_+) + \lambda D(u^* - u_-).\end{aligned}$$

Thus, if $u^* - u^0$, $u^* - u_-$, $u^* - u_+$ all had the same sign, we would get a contradiction. Thus, at least two of the three terms must have opposite signs, which implies (2.3.30).

□

2.4 Convergence to the entropy solution

In this section, we restrict our attention to implicit monotone discretizations corresponding to the conservation law

$$u_t + f(u)_x = 0, \qquad (x,t) \in \mathbb{R} \times \mathbb{R}^+, \tag{2.4.1}$$

i.e. when $\phi(x) \equiv 1$ is constant and the flux function does not vary in space (but is generally non-convex and/or non-monotonic). Kružkov [45] has shown that (2.4.1) has a unique entropy-satisfying weak solution, as stated in the following theorem.

Theorem 2.11 (Kružkov). *If f is locally Lipschitz continuous, then for any $u_0 \in BV(\mathbb{R})$ and for any $T > 0$ there is a unique $u \in BV(\mathbb{R} \times [0,T]) \cap C^0([0,T], L^1_{loc}(\mathbb{R}))$ such that u is a weak solution, i.e.*

$$\iint_{\mathbb{R}\times[0,T]} (u\psi_t + f(u)\psi_x) dx\, dt + \int_{\mathbb{R}} u_0(x)\psi(x,0)dx = 0 \tag{2.4.2}$$

for all $\psi \in C_0^\infty(\mathbb{R} \times [0,T])$ and, in addition, satisfies the entropy condition: For all $\psi \in C_0^\infty(R \times [0,T])$ with $\psi \geq 0$, and for all $c \in \mathbb{R}$,

$$\iint_{\mathbb{R}\times[0,T]} [|u-c|\psi_t + \operatorname{sgn}(u-c)(f(u)-f(c))\psi_x]\, dx\, dt \geq 0. \tag{2.4.3}$$

The classical approach for establishing convergence to the unique entropy solution proceeds as follows (cf. [24, 41, 70]):

1. Show that the sequence of numerical approximations remains uniformly bounded and has uniformly bounded total variation as $\Delta x, \Delta t \to 0$. This ensures the set of numerical approximations is precompact in $L^1_{\text{loc}}(\mathbb{R} \times [0, T])$, which allows one to produce a convergent subsequence;

2. Show that the numerical flux is consistent and satisfies a discrete entropy inequality. By the Lax-Wendroff theorem [46], this implies the limit u of the convergent subsequence satisfies (2.4.2) and (2.4.3) in Theorem 2.11;

3. Verify that the entropy-satisfying weak solution is unique. In the 1D scalar case this is a result of Theorem 2.11. This ensures all subsequences have the same limit point, so that the finite difference scheme is convergent as $\Delta x, \Delta t \to 0$.

A detailed argument along the above lines can be found in [24, 50, 70] and will not be repeated here. Instead we focus on checking the various criteria listed above. The numerical flux is assumed to be consistent, and by Theorem 2.6, the discrete solution is uniformly bounded for spatial and temporal grid size. Thus, we only need to verify that the numerical approximations have bounded total variation, and that a discrete entropy inequality exists. The following two lemmas address these questions.

Lemma 2.12. *Assume the hypotheses of Theorem 2.6, and suppose for $n \geq 1$ the discrete solution $\{u_i^n\}_{i=-\infty}^{\infty}$ is given by the unique bounded solution satisfying* (2.3.1). *Assume the initial data $\{u_i^0\}_{i=-\infty}^{\infty}$ has bounded total variation, i.e.*

$$TV(u^0) := \sum_{i=-\infty}^{\infty} |u_{i+1}^0 - u_i^0| < \infty.$$

Then $TV(u^n) < \infty$ for all $n \geq 1$, and

$$TV(u^{n+1}) \leq TV(u^n) \qquad \textit{for all } n.$$

Proof. For notational simplicity we write $u_i = u_i^{n+1}$, $v_i = u_i^n$, $\Delta u_{i+1/2} = u_{i+1} - u_i$. By a manipulation similar to the one in Proposition 2.10, we have, for each $i \in \mathbb{Z}$,

$$u_i - v_i - \lambda C_i \Delta u_{i+1/2} + \lambda D_i \Delta u_{i-1/2} = 0, \tag{2.4.4}$$

where

$$\begin{aligned} C_i &= \frac{F(u_i, u_i) - F(u_i, u_{i+1})}{u_{i+1} - u_i} \geq 0, \\ D_i &= \frac{F(u_i, u_i) - F(u_{i-1}, u_i)}{u_i - u_{i-1}} \geq 0. \end{aligned}$$

Writing (2.4.4) for cells i, $i+1$ and subtracting gives

$$\begin{aligned} \Delta u_{i+1/2} - \Delta v_{i+1/2} - \lambda C_{i+1} \Delta u_{i+3/2} \\ + \lambda D_{i+1} \Delta u_{i+1/2} + \lambda C_i \Delta u_{i+1/2} - D_i \Delta u_{i-1/2} = 0. \end{aligned}$$

Rearrange and get

$$(1 + \lambda C_i + \lambda D_{i+1}) \Delta u_{i+1/2} = \Delta v_{i+1/2} + \lambda C_{i+1} \Delta u_{i+3/2} + \lambda D_i \Delta u_{i-1/2}.$$

Since $C_i, C_{i+1}, D_i, D_{i+1}$ are all non-negative, the triangle inequality gives

$$(1 + \lambda C_i + \lambda D_{i+1}) |\Delta u_{i+1/2}| \leq |\Delta v_{i+1/2}| + \lambda C_{i+1} |\Delta u_{i+3/2}| + \lambda D_i |\Delta u_{i-1/2}|. \tag{2.4.5}$$

Summing (2.4.5) for i from $-N$ to N and making some cancellations, we get

$$\begin{aligned} \sum_{i=-N}^{N} |\Delta u_{i+1/2}| &\leq \sum_{i=-N}^{N} |\Delta v_{i+1/2}| + \lambda C_{N+1} |\Delta u_{N+3/2}| \\ &\quad - \lambda C_{-N} |\Delta u_{-N+1/2}| + \lambda D_{-N} |\Delta u_{-N-1/2}| - \lambda D_{N+1} |\Delta u_{N+1/2}| \\ &\leq \sum_{i=-N}^{N} |\Delta v_{i+1/2}| + \lambda C_{N+1} |\Delta u_{N+3/2}| + \lambda D_{-N} |\Delta u_{-N-1/2}| \\ &\leq TV(v) + 4\lambda K_B \cdot B, \end{aligned}$$

where $\|u^0\|_\infty < B$ and K_B is the local Lipschitz constant. Since the last expression is finite and independent of N, the monotone convergence theorem guarantees that if we let N approach infinity, the series will converge, so we get $TV(u) < \infty$. Moreover, it implies that for every $\varepsilon > 0$ there exists N (which depends on ε) such that

$$\sum_{|i|>N} |\Delta u_{i+1/2}| \leq \frac{1}{2} \min\{\, \varepsilon, \frac{\varepsilon}{\lambda K_B}\}.$$

Thus, we have

$$\begin{aligned}
TV(u) &\leq \frac{\varepsilon}{2} + \sum_{i=-N}^{N} |\Delta u_{i+1/2}| \\
&\leq \frac{\varepsilon}{2} + \sum_{i=-N}^{N} |\Delta v_{i+1/2}| + \lambda C_{N+1} |\Delta u_{N+3/2}| + \lambda D_{-N} |\Delta u_{-N-1/2}| \\
&\leq \frac{\varepsilon}{2} + TV(v) + \lambda K_B (|\Delta u_{N+3/2}| + |\Delta u_{-N-1/2}|) \\
&\leq \frac{\varepsilon}{2} + TV(v) + \lambda K_B \sum_{|i|>N} |\Delta u_{i+1/2}| \\
&\leq \frac{\varepsilon}{2} + TV(v) + \lambda K_B \cdot \frac{\varepsilon}{2\lambda K_B} \\
&\leq TV(v) + \varepsilon.
\end{aligned}$$

Finally, letting $\varepsilon \to 0$ gives $TV(u) \leq TV(v)$, as required. □

For the next lemma, recall that if u is an entropy-satisfying weak solution to (2.4.1), it must satisfy an entropy inequality of the form

$$\varphi(u)_t + \psi(u)_x \leq 0 \tag{2.4.6}$$

in the weak sense, where $\varphi(\cdot)$ is an arbitrary C^2 function with $\varphi'' > 0$, and (φ, ψ) are related by $\psi' = \varphi' f'$. Kružkov showed in [45] that this formulation is equivalent to requiring that condition (2.4.3) be satisfied for all $c \in \mathbb{R}$.

Lemma 2.13. *Assume the hypotheses of Theorem 2.6, and let $\{u_i^n\}$ be the unique* *bounded solution satisfying* (2.3.1)*. Let* (φ, ψ) *be an entropy/flux pair. Then there exist functions* $\Phi = \Phi(u)$ *and* $\Psi = \Psi(u^-, u^+)$ *that are consistent with* (φ, ψ) *(i.e.* $\Phi(u) = \varphi(u)$ *and* $\Psi(u, u) = \psi(u)$*) such that* (Φ, Ψ) *satisfies a* discrete entropy inequality*:*

$$\Phi(u_i^{n+1}) - \Phi(u_i^n) + \lambda \left[\Psi(u_i^{n+1}, u_{i+1}^{n+1}) - \Psi(u_{i-1}^{n+1}, u_i^{n+1})\right] \leq 0. \tag{2.4.7}$$

Proof. The development essentially follows [75]. Define the *entropy variables*

$$v := \varphi'(u).$$

Since $\varphi'' > 0$, φ' is one-to-one, so we can do a change of variables and let $u = u(v)$. So we can define the potential function

$$q(v) = \int_0^v f(u(\eta))d\eta,$$

which is used to define the discrete entropy/flux pair:

$$\begin{aligned} \Phi(u) &:= \varphi(u) \\ \Psi(u_i, u_{i+1}) &:= \frac{1}{2}(v_i + v_{i+1})F_{i+1/2} - \frac{1}{2}(q(v_i) + q(v_{i+1})). \end{aligned}$$

It is easily seen that Ψ is consistent with ψ (by showing that $\frac{d}{du}\Psi(u,u) = \varphi' f'$). Then since $\varphi = \Phi$ is convex, we have

$$\Phi(u_i^{n+1}) + \Phi'(u_i^{n+1})(u_i^n - u_i^{n+1}) \leq \Phi(u_i^n),$$

so that

$$\begin{aligned} 0 &\geq \Phi(u_i^{n+1}) - \Phi(u_i^n) + v_i^{n+1}(u_i^n - u_i^{n+1}) \\ 0 &\geq \Phi(u_i^{n+1}) - \Phi(u_i^n) + \lambda v_i^{n+1}(F_{i+1/2}^{n+1} - F_{i-1/2}^{n+1}) \\ 0 &\geq \Phi(u_i^{n+1}) - \Phi(u_i^n) + \lambda \left[\left(v_i^{n+1}F_{i+1/2}^{n+1} + q(v_i^{n+1})\right) - \left(v_i^{n+1}F_{i-1/2}^{n+1} + q(v_i^{n+1})\right)\right] \end{aligned}$$

So for (2.4.7) to hold it is sufficient demonstrate the following inequalities:

$$\Psi(u_i, u_{i+1}) \leq v_i F_{i+1/2} - q(v_i), \tag{2.4.8}$$

$$\Psi(u_{i-1}, u_i) \geq v_i F_{i+1/2} - q(v_i). \tag{2.4.9}$$

From the definition of Ψ we get

$$\begin{aligned}
&\Psi(u_i, u_{i+1}) - v_i F_{i+1/2} + q(v_i) \\
&\quad = \frac{1}{2}(v_i + v_{i+1})F_{i+1/2} - \frac{1}{2}(q(v_i) + q(v_{i+1})) - v_i F_{i+1/2} + q(v_i) \\
&\quad = \frac{1}{2}(v_{i+1} - v_i)F_{i+1/2} - \frac{1}{2}(q(v_{i+1}) - q(v_i)) \\
&\quad = \frac{1}{2}\int_{v_i}^{v_{i+1}} \big[F_{i+1/2} - f(u(\eta))\big]\, d\eta \\
&\quad = \frac{1}{2}\int_{v_i}^{v_{i+1}} [F(u_i, u_{i+1}) - F(u(\eta), u(\eta))]\, d\eta \\
&\quad = \frac{1}{2}\int_{v_i}^{v_{i+1}} \big[\big(F(u_i, u_{i+1}) - F(u_i, u(\eta))\big) + \big(F(u_i, u(\eta)) - F(u(\eta), u(\eta))\big)\big]\, d\eta,
\end{aligned}$$

where η lies between v_i and v_{i+1}. If $v_i \leq v_{i+1}$, then $u_i \leq u(\eta) \leq u_{i+1}$, so that

$$\begin{aligned}
F(u_i, u_{i+1}) - F(u_i, u(\eta)) &\leq 0 \\
F(u_i, u(\eta)) - F(u(\eta), u(\eta)) &\leq 0,
\end{aligned}$$

and hence the integrand is non-positive. Analogously, $v_i \geq v_{i+1}$ implies that the integrand is non-negative, so either way the integral cannot be positive, thus proving (2.4.8). Relation (2.4.9) is proved similarly, and the lemma follows. □

2.5 Accuracy of phase-based upstreamed solutions

In this section, we investigate the accuracy of the numerical solution obtained from phase-based upstreaming when we vary the spatial and temporal grid. Our test case consists of the 1D countercurrent flow problem (§2.2.1), with domain $\Omega = [0, 1]$. Water is injected at the boundary $x_D = 0$ and a pressure boundary condition is maintained at $x_D = 1$. The hyperbolic form of the problem is described by

$$\frac{\partial S}{\partial t_D} + \frac{\partial f(S)}{\partial x_D} = 0.$$

The flux function $f(S)$ is shown in Figure 1(b), with a sonic point at $S = 0.49$; countercurrent flow occurs whenever $S \geq 0.49$. The initial saturation profile is a step function with

$$S^0(x_D) = \begin{cases} 1, & 0 \leq x_D < 0.2 \\ 0 & 0.2 < x_D \leq 1. \end{cases}$$

The numerical solution is compared with the analytical solution at time $t_D = 0.15$. Because of the sonic point, the solution contains two shocks connected by a rarefaction; one shock moves to the right with a velocity of 3.9, and the other travels to the left with a velocity of -1.2. When considering the accuracy of a numerical solution, two error measures are shown:

- The L^1-*error*, which is the difference between the numerical and the analytical solution in the L^1-norm;

- The *front dispersion*, which is the distance between analytical shock front and the leftmost point for which the numerical solution becomes zero.

We also measure how difficult the nonlinear problem is by showing, for each test case, the average number of nonlinear Gauss-Seidel iterations required to converge each time step. We remark that this measure is only useful for problems with countercurrent flow, since Gauss-Seidel always converges in one iteration in the cocurrent case (cf. section 2.3.6).

2.5.1 Refinement under fixed mesh ratio

Here we refine the grid under a fixed mesh ratio $\Delta x/\Delta t$, which in turn yields a fixed CFL number of 4.10, which is above the CFL limit for explicit schemes. Figure 2.3 shows the plots for $N = 25, 50, 100, 200, 400$, and Table 2.1 shows the L^1-error and front dispersion data. The plots show that the numerical solution converges to the

Table 2.1: Accuracy of numerical solutions for a fixed CFL number.

N	$t_D/\Delta t$	CFL	L^1-error	Front dispersion	Average # iterations
25	5	4.10	0.0889	> 0.215	5.2
50	10	4.10	0.0665	> 0.215	4.9
100	20	4.10	0.0444	0.116	4.4
200	40	4.10	0.0273	0.066	4.2
400	80	4.10	0.0168	0.039	4.1

analytical solution even though the CFL number is greater than 1, which confirms our analysis. Moreover, both the L^1 error and the front dispersion are converging a bit worse than linearly, with a ratio of about 0.61 and about 0.58 respectively for every refinement by a factor of two. Also note the poor resolution near the left boundary $N = 25, 50, 100$, where instead of approaching $S = 1$, the solution is closer to $S_c \approx 0.27$ at the left boundary. For these coarser grids, the numerical solution has a hard time deciding whether the left-moving wave has reached the boundary, which is maintained at $S(x = 0) = S_c$ (see Equation (2.2.10)). For higher resolutions ($N = 200, 400$), the artifact has disappeared and the numerical solution reproduces the back end of the saturation profile quite accurately. The average number of Gauss-Seidel iterations required for convergence are all similar, so refining the grid for a fixed mesh ratio does not increase the difficulty of the problem for the nonlinear solver.

2.5.2 Spatial refinement for fixed time steps

Here, we refine the spatial grid only while fixing the time-step size. Figure 2.4 and Table 2.2 show the results for $N = 25, 50, 100, 200, 400$, and a time-step size of $\Delta t_D = 0.0075$, i.e. we use 20 time steps to integrate up to $t_D = 0.15$. We see that even though the $N = 25$ case has a CFL number close to 1, the grid is clearly too coarse, and the shock front is very poorly resolved. The accuracy increases substantially when the spatial grid is refined to $N = 50, 100$, even though the CFL number becomes progressively larger; thus, the CFL number by itself is not a good measure of solution quality. However, the improvement due to spatial grid refinement becomes negligible for $N > 100$, since time discretization is now the dominant source of

Table 2.2: Accuracy of numerical solutions for a fixed time step size.

N	$t_D/\Delta t$	CFL	L^1-error	Front dispersion	Average # iterations
25	20	1.02	0.0673	> 0.215	2.6
50	20	2.05	0.0529	0.156	3.3
100	20	4.10	0.0444	0.116	4.4
200	20	8.20	0.0378	0.101	6.4
400	20	16.40	0.0366	0.094	9.2

error. In addition, the average number of iterations required to attain convergence increases with each refinement: as we refine the grid, we are solving increasingly difficult problems, even though the improvement in solution accuracy will stagnate beyond a certain point. Thus, even though the fully-implicit method can tolerate arbitrarily large CFL numbers, one should not hope to improve solution accuracy indefinitely simply by using a finer spatial grid, without making a corresponding reduction of time-step size.

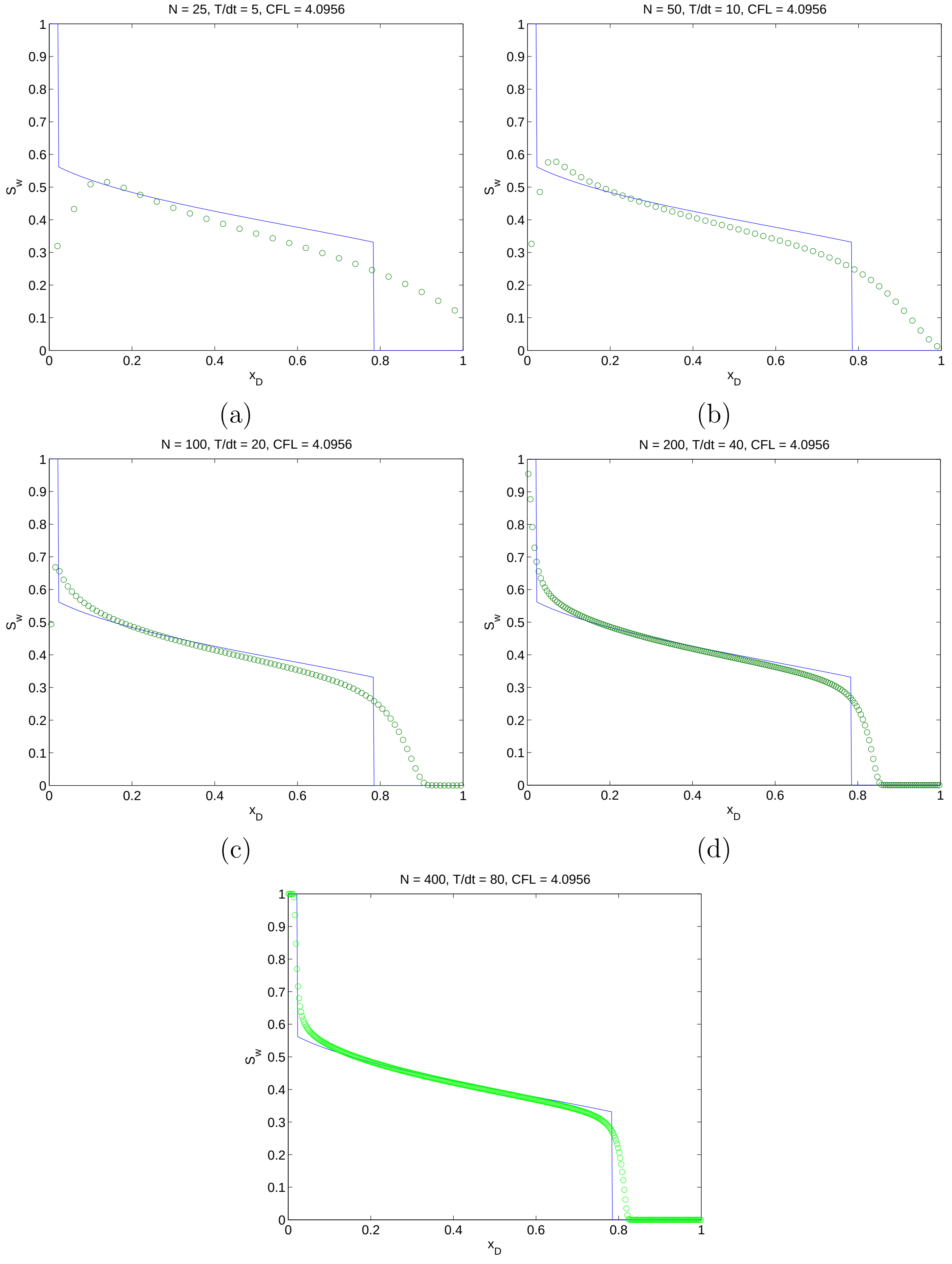
2.5.3 Non-uniform grids

The real advantage of the fully-implicit method over an explicit scheme lies in its efficiency when applied to a heterogeneous problem, where the porosity $\phi(x)$ and permeability $K(x)$ can vary by orders of magnitude over the domain. In these problems, the CFL condition is determined by the minimum porosity in the domain, which can be much smaller than the average porosity. To illustrate this point, we show an example in which the spatial grid is non-uniform (which, based on Remark 2.3.7, is equivalent to the spatially-varying porosity case). The non-uniform grid contains 50 gridblocks, with $\Delta x_{\max}/\Delta x_{\min} = 96$. Figure 2.5 and Table 2.3 compare the numerical solutions obtained from this grid to the uniform-grid solutions. We see that the solutions are qualitatively (from the plots) and quantitatively (from the L^1-error and front dispersion) not very different from their uniform counterparts, even though the CFL number is 50 times larger in the non-uniform case. Thus, an explicit integrator would have to take unacceptably small time steps, whereas an implicit method allows time steps that are much more reasonable. In addition, the average number

Table 2.3: Accuracy of numerical solutions for a non-uniform grid.

	N	$t_D/\Delta t$	CFL	L^1-error	Front dispersion	Avg. # its
Non-uniform	50	20	105.80	0.0566	0.180	3.2
	50	50	42.30	0.0475	0.132	2.2
Uniform	50	20	2.05	0.0529	0.156	3.3
	50	50	0.82	0.0435	0.116	2.1

of iterations required for Gauss-Seidel convergence is roughly the same for both the uniform and non-uniform case, so the resulting nonlinear equations are not harder to solve, despite the large CFL numbers.



(a)

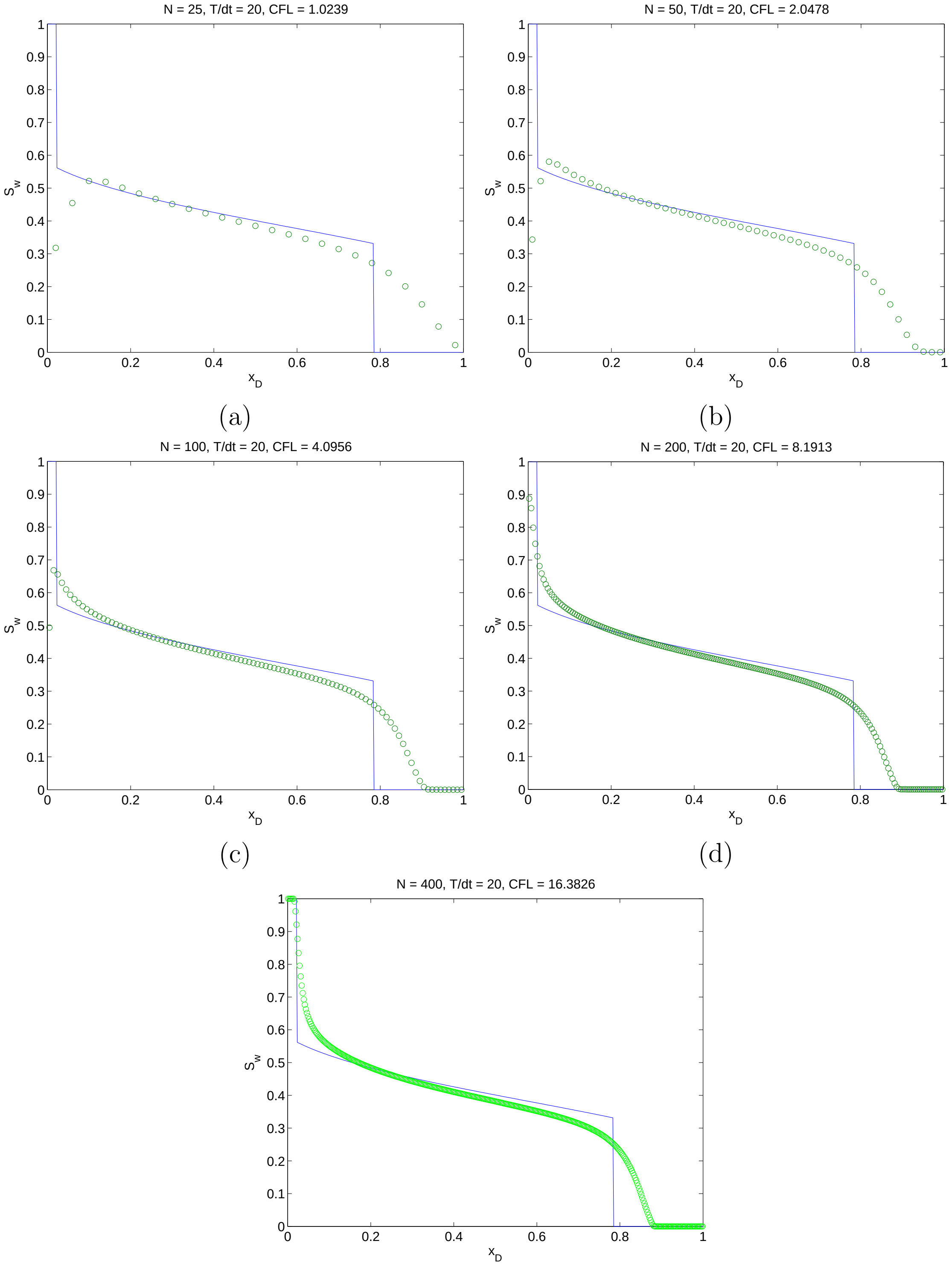
(b)

(c)

(d)

(e)

Figure 2.3: Numerical solution at different resolutions, CFL = 4.10, $t_D = 0.15$.



(a)

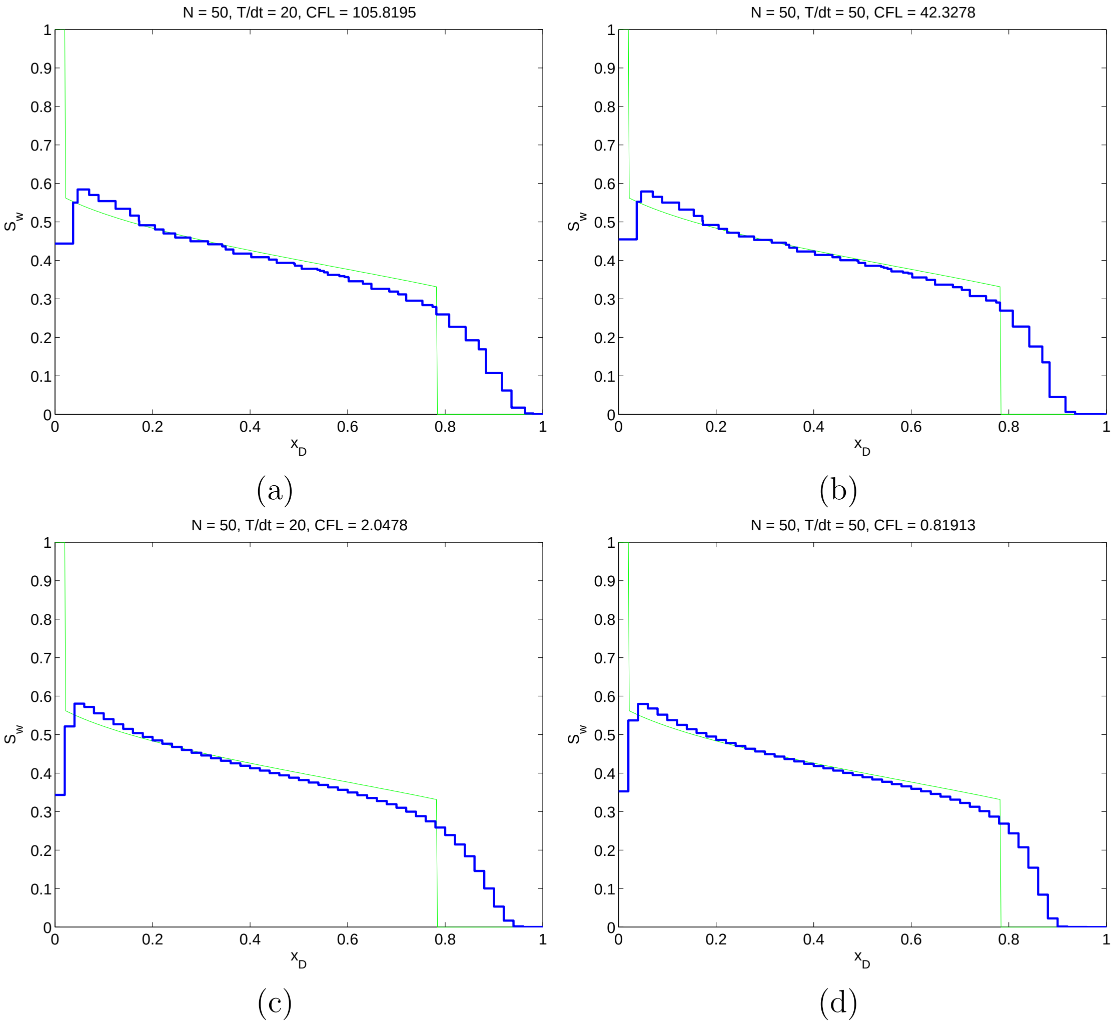
(b)

(c)

(d)

(e)

Figure 2.4: Numerical solution for different spatial grids, 20 time steps, $t_D = 0.15$.



(a) (b)

(c) (d)

Figure 2.5: Numerical solutions obtained from a non-uniform grid ((a) and (b)) and their uniform-grid counterparts ((c) and (d)), $t_D = 0.15$.

Chapter 3

Potential Ordering

The main theme of this thesis is the reordering of equations and variables in a way that allows a partial decoupling of the problem into a sequence of single-cell problems that are easier to solve. The basic insight is to perform reordering based on flow direction information, which is provided by the pressure field. This approach is intuitive because saturation information travels from upstream to downstream, so one expects methods that respect this ordering to be more efficient than methods that are blind to upstream information. We have already seen in section 2.3.6 that in the 1D cocurrent case, nonlinear Gauss-Seidel converges in exactly one iteration if, and only if, the cells are ordered from upstream to downstream. Thus, ordering can have a large impact on the performance of solution algorithms.

For a problem with n_p phases, there are n_p equations and unknowns associated with each block, which means there are multiple ways of ordering these equations while respecting the direction of flow. We can distinguish between the following two categories of ordering:

1. Cell-based ordering, in which all the equations and variables aligned with a cell (control volume) are grouped together as a block, and reordering only applies at the cell level;

2. Phase-based ordering, in which all the equations and variables corresponding to a particular phase p are grouped together, and a different cell ordering can be used for each block.

The two approaches are useful in different situations and they both contribute to the various nonlinear solvers and preconditioning algorithms presented in subsequent chapters.

3.1 Methods derived from cell-based ordering

In cell-based approaches, the cells are ordered along the flow direction (based on either the total velocity field or the pressure field of the "dominant" phase). The single-cell problems thus obtained are generally n_p-by-n_p systems of nonlinear equations corresponding to local mass balances, one per component. Decoupling occurs by assuming that the inward fluxes from upstream cells are known, and that downstream dependence is weak enough so that the single-cell solution will not be significantly affected. These approaches work well for cocurrent flow problems because downstream dependence is effectively nil in such cases, and there is no ambiguity in the ordering since all phases flow in the same direction.

Cascade method

The Cascade method was proposed by Appleyard and Cheshire [4] as an acceleration scheme for the basic Newton method. A brief description of the method follows. Suppose we have an n_p-phase model (n_p= 2 or 3) in which we discretize the domain into N gridblocks. The first step in the Cascade method is the same as in the ordinary Newton method: namely, we linearize the n_pN conservation equations and solve the n_pN-by-n_pN linear system $J(x^{(\nu)})\delta x^{(\nu)} = -R^{(\nu)}(x^{(\nu)})$ for $\delta x^{(\nu)}$. Next, we apply a linear update to pressure variables P_o only, leaving the saturations intact for the time being. Using this new pressure field, we update the potential for each phase, and then we order the cells from the highest potential to the lowest. This is the order in which the Cascade sweep should be performed. Note that there is a choice in the ordering because the potential sequence can be different for each phase. Appleyard and Cheshire suggest that one Cascade sweep be done for the potential sequence of each phase, although the method was only demonstrated for a two-phase flow problem.

```
1 Form the full Jacobian $J$, evaluated at $(S^k, P_o^k)$ ;
2 Solve $J \begin{bmatrix} \delta S^k \\ \delta P_o^k \end{bmatrix} = r^k$ ;
3 Compute $P_o^{k+1} = P_o^k + \delta P_o^k$ ;
4 Reorder the cells so that $P_{o,i} \geq P_{o,j}$ whenever $i > j$ ;
5 For $i = 1, \ldots, N$ :
6     Solve (3.1.1) at cell $i$ for $S_{w,i}$ and $P_{o,i}$ ;
7     Update $S_{w,i}^{k+1}$ using the value from line 6 ;
8     Compute outward fluxes $FO_p(S_w, P_o)$ for subsequent $i$ ;
9 end for
```

Figure 3.1: One iteration of the Cascade method [4].

Each Cascade sweep requires the solution of N single-cell problems, where N is the number of cells in the grid. For a two-phase problem, a single-cell problem has the form

$$\begin{aligned} f_o(S_w, P_o) &= \frac{1}{\Delta t}\Delta M_o(S_w, P_o) + FO_o(S_w, P_o) - FI_o - q_o = 0 \\ f_w(S_w, P_o) &= \frac{1}{\Delta t}\Delta M_w(S_w, P_o) + FO_w(S_w, P_o) - FI_w - q_w = 0, \end{aligned} \tag{3.1.1}$$

where ΔM_p is the accumulation of phase p, FO_p and FI_p are the outward and inward fluxes of phase p respectively, and q_p are the well terms. For a three-phase problem, we would have three such equations. We assume that the inward fluxes are known and independent of the values of S_w and P_o at the cell, which is valid provided that all neighboring cells at a higher potential have been processed, and there is no countercurrent flow. We now have a system of two nonlinear equations in two unknowns, which can be efficiently solved for S_w and P_o. The computed S_w are taken to be the saturation solution for the nonlinear iteration, and the computed FO_p are used as the influx for subsequent single-cell problems. The computed P_o, on the other hand, are discarded, since their only purpose is to ensure local mass conservation for both phases and do not yield an accurate approximation for the global pressure field. In other words, the approximate solution $(S^{(\nu)}, P_o^{(\nu)})$ takes its saturation values from the single-cell problems, but the pressure values are obtained from the linear update. Figure 3.1 outlines one step of the cascade method.

Consider a one-dimensional model problem with

- incompressible flow,
- an injection boundary condition on the left,
- a pressure boundary condition on the right, and
- no countercurrent flow (e.g., horizontal reservoir with no capillarity).

It can be shown that the Cascade method converges to the solution in two iterations for this problem (see Appendix C for a proof). However, this ceases to be true in the presence of countercurrent flow or in multiple dimensions. Also, the formulation may break down if the phase potential chosen to order the cells contains local minima; in this case, the cell whose potential is at a local minimum will lack an outward flux term FO_p, so it would be impossible to satisfy mass balance for both phases no matter what S_w and P_o are. This is an important drawback because in practical applications it is usually impossible to guarantee the absence of local minima in the pressure field when the solution has not converged, especially when the initial guess is poor.

The Natvig approach

Natvig, Lie and Eikemo [56] proposed a cell-based reordering method for solving the multiphase advection problem in the absence of gravity and capillarity. In [56] the reordering was applied to equations obtained from a discontinuous Galerkin discretization, but it can equally be applied to the standard finite volume methods described in section 1.2.1. Basically, a topological sort (cf. [22]) is performed on the directed acyclic graph $G = (V, E)$, whose nodes V are the control volumes, and whose edges E are the directions of the total velocity across cell interfaces (which coincide with the flow directions for each phase, since there is no countercurrent flow). The single-cell problems, each consisting of an n_p-by-n_p nonlinear system, are solved in the topological order from upstream to downstream by Newton's method, for example. Since the pressure and total velocity fields are regarded as part of the data rather than the unknowns, this ordering completely decouples the system, just like Gauss-Seidel is exact for cocurrent 1D flow. In fact, this approach can be regarded as a block nonlinear Gauss-Seidel method, which is exact as long as the nonlinear system is block lower triangular. Again, convergence is no longer superlinear when countercurrent flow is present, and a robust implementation in the block case becomes non-trivial (see [28] for a discussion).

3.2 Phase-based ordering

In this section, we present an ordering of equations and unknowns that allows us to solve for saturation one unknown at a time, even in multiple dimensions and/or in the presence of gravity and countercurrent flow. First, we explain how to construct this ordering in the absence of countercurrent flow; in this case we recover the Appleyard and Cheshire Cascade ordering [4]. We then extend the ordering to treat countercurrent flow due to gravity, and finally we show how to deal with capillarity.

3.2.1 Cocurrent flow

Consider the two-phase model outlined in section 1.1. In the absence of gravity and capillary forces, all phases will be flowing in the same direction, which is given by the negative pressure gradient $-\nabla p$ (i.e., from high to low pressure). Thus, in the finite volume discretization, the flux term between cells i and l,

$$F_{il} = \begin{cases} K \cdot \dfrac{k_{rp}(S_l)}{\mu_p} \dfrac{p_l - p_i}{\Delta x}, & p_l \geq p_i \\ K \cdot \dfrac{k_{rp}(S_i)}{\mu_p} \dfrac{p_l - p_i}{\Delta x}, & p_l \leq p_i \end{cases} \tag{3.2.1}$$

depends only on the saturation of the upstream cell. Suppose we reorder the cells such that they appear in decreasing order of pressure, i.e. $p_i \geq p_j$ whenever $i < j$. Then for all j, the component conservation equations for cell j depend only on saturations S_i with $i \leq j$. Thus, we can rearrange the system of nonlinear equations to the form

$$\begin{aligned} f_{c1}(S_1, \qquad\qquad\quad & p_1, \dots, p_N) = 0 \\ f_{c2}(S_1, S_2, \qquad\quad & p_1, \dots, p_N) = 0 \\ & \vdots \\ f_{cN}(S_1, S_2, \dots, S_N, & p_1, \dots, p_N) = 0, \end{aligned} \tag{3.2.2}$$

where $c = o, w$ are the oil and water components, respectively. Notice how the saturation part of the equations becomes "triangular". Thus, if we have the exact pressure solution $p_1, \ldots, p_N$, we can perform a "forward substitution" and solve a series of single-variable nonlinear equations to obtain the saturations $S_1, \ldots, S_N$. We remark that the triangularity carries over to the Jacobian matrix, which now has the form

$$J = \begin{array}{cc} S_w & p \\ \left[\begin{array}{c} J_{ww} \\ J_{ow} \end{array}\right. & \left.\begin{array}{c} J_{wp} \\ J_{op} \end{array}\right] \end{array} \begin{array}{l} \\ \text{water equation} \\ \text{oil equation} \end{array} \tag{3.2.3}$$

where J_{ww} is lower triangular.

In the three-phase case, we have two saturation variables per cell, which we can choose as S_w and S_o without loss of generality. Since the black oil model assumes that k_{rw} depends solely on S_w, the above construction can be used to order the water equations. Now k_{ro} depends on both S_w and S_o, but we can maintain triangularity by writing all the water equations first before writing the oil and gas equations. The nonlinear system then looks like

$$\begin{aligned}
& f_{w1}\,(S_{w1}, && p_1, \ldots, p_N) = 0 \\
& f_{w2}\,(S_{w1}, S_{w2}, && p_1, \ldots, p_N) = 0 \\
& \qquad\qquad\vdots \\
& f_{wN}(S_{w1}, \ldots, S_{wN}, && p_1, \ldots, p_N) = 0 \\
& f_{o1}\,(S_{w1}, \ldots, S_{wN}, S_{o1}, && p_1, \ldots, p_N) = 0 \\
& \qquad\qquad\vdots \\
& f_{oN}\,(S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}, p_1, \ldots, p_N) = 0 \\
\text{and}\quad & f_{gi}\,(S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}, p_1, \ldots, p_N) = 0, \quad i = 1, \ldots, N.
\end{aligned} \tag{3.2.4}$$

In this case the corresponding Jacobian would have the form

$$J = \left[\begin{array}{cc|c|l} S_w & S_o & p & \\ J_{ww} & & J_{wp} & \text{water equation} \\ J_{ow} & J_{oo} & J_{op} & \text{oil equation} \\ \hline J_{gw} & J_{go} & J_{gp} & \text{gas equation} \end{array}\right] \tag{3.2.5}$$

with J_{ww} and J_{oo} lower triangular, which implies the entire upper-left block is lower triangular. Note that J_{ow} will also be lower triangular, since all phases have the same upstream direction. However, this fact is not needed to justify solving for S_w and S_o using forward substitution.

3.2.2 Countercurrent flow due to gravity

In the presence of gravity, buoyancy forces can cause different phases to flow in opposite directions. The upstream direction for each phase p is determined by the sign of $(\Phi_{p,i} - \Phi_{p,l})$, where

$$\Phi_{p,i} = p_i - \gamma_p z_i \tag{3.2.6}$$

is the phase potential at cell i, z_i is the depth of the cell, and γ_p is the specific gravity of phase p. Despite possible differences in upstream directions, we are interested in maintaining the triangular forms shown in (3.2.2) and (3.2.4) (and equivalently (3.2.3) and (3.2.5)). For two-phase flow, one can simply use Φ_w for ordering, since one only needs J_{ww} (and not J_{ow}) to be triangular. For three-phase flow, we need both J_{ww} and J_{oo} to be lower triangular. Clearly, no cell-based ordering can accomplish this; we need to order the water and oil phases separately. The trick is to exploit the relative permeability dependencies (1.1.9) in such a way that triangularity is preserved.

Unlike the cocurrent flow case, we can no longer align the ordering of equations and variables with cell ordering. Thus, in the sequel, subscripts (such as k in $\Phi_{p,k}$) always denote the value of the scalar field (in this case, the potential of phase p) at cell k *in the natural ordering.* This is because we concentrate on ordering the equations and unknowns, rather than the cells themselves.

Let $\sigma_1, \ldots, \sigma_N$ and $\tau_1, \ldots, \tau_N$ be permutations such that

$$\Phi_{w,\sigma_i} \geq \Phi_{w,\sigma_j} \qquad \text{whenever } i < j, \tag{3.2.7}$$

$$\Phi_{o,\tau_i} \geq \Phi_{o,\tau_j} \qquad \text{whenever } i < j. \tag{3.2.8}$$

In other words, if cell k is such that $\Phi_{w,k} > \Phi_{w,l}$ for any other l, then $\sigma_1 := k$. Suppose we order first all the water equations and the associated variables S_w using the σ ordering, and then order the oil equations and the associated variables S_o using the τ ordering. The nonlinear system then looks like

$$\begin{aligned}
&f_{w,\sigma_1}\,(S_{w,\sigma_1}, && p_1, \ldots, p_N) = 0\\
&f_{w,\sigma_2}\,(S_{w,\sigma_1}, S_{w,\sigma_2}, && p_1, \ldots, p_N) = 0\\
&\qquad\qquad\vdots\\
&f_{w,\sigma_N}(S_{w,\sigma_1}, \ldots, S_{w,\sigma_N}, && p_1, \ldots, p_N) = 0\\
&f_{o,\tau_1}\;\,(S_{w,\sigma_1}, \ldots, S_{w,\sigma_N}, S_{o,\tau_1}, && p_1, \ldots, p_N) = 0\\
&\qquad\qquad\vdots\\
&f_{o,\tau_N}\,(S_{w,\sigma_1}, \ldots, S_{w,\sigma_N}, S_{o,\tau_1}, \ldots, S_{o,\tau_N}, && p_1, \ldots, p_N) = 0\\
\text{and}\quad &f_{gi}\quad(S_{w,\sigma_1}, \ldots, S_{w,\sigma_N}, S_{o,\tau_1}, \ldots, S_{o,\tau_N}, && p_1, \ldots, p_N) = 0, \quad i = 1, \ldots, N.
\end{aligned} \tag{3.2.9}$$

Now consider the pattern of the corresponding Jacobian matrix. Clearly, J_{ww} is still lower triangular because of (3.2.7), and J_{oo} is lower triangular because of (3.2.8). The only effect of countercurrent flow is that J_{ow} will no longer be lower triangular, because the S_w are not arranged in the order of decreasing oil potential, Φ_o. However, as long as the upper-left 2×2 block in (3.2.5) is lower triangular, we can use forward substitution to solve for S_w and S_o once the pressures are known.

3.2.3 Capillarity

So far, in the absence of capillary effects, the saturation dependence in each equation is purely upstream; thus, for a given phase, saturations downstream from cell i do not appear in equation i. When capillary effects are present, equation i involves phase pressures from all neighboring cells, be they upstream or downstream from cell i. In the standard approach, we can only choose one phase pressure as a primary variable; the other phase pressures must be expressed as

$$p_q = p_p + P_{cpq}(S), \tag{3.2.10}$$

where p_p is the primary phase pressure and p_q is the pressure of another phase. Thus, when capillarity is present, we must choose our primary variables carefully to avoid introducing downstream dependence on saturation that cannot be removed by simply reordering the equations and unknowns. Choosing p_w, the water-phase pressure, as the primary pressure variable allows us to maintain the triangularity in the upper-left block of (3.2.5). Note that choosing p_g causes the water equations to depend on S_o, since $p_w = p_g - P_{cog}(S_g) - P_{cow}(S_w)$ and $S_g = 1 - S_w - S_o$. This would completely destroy the triangularity of the block. If we instead choose p_o, then there would be no S_o dependence, but there would be both upstream and downstream dependence on S_w due to $p_w = p_o - P_{cow}(S_w)$, which is undesirable. Thus, the only choice that leaves the water equation intact (i.e., a triangular J_{ww}) is p_w.

We need to ensure that J_{oo} is still lower triangular when p_w is used. We have

$$p_o = p_w + P_{cow}(S_w), \tag{3.2.11}$$

which means we introduce downstream dependence on S_w, but not on S_o. Hence, the J_{ow} block will now contain downstream terms, but the J_{oo} block remains unchanged. Thus, the upper-left block remains triangular, as before. The same analysis carries over to the nonlinear equations (3.2.9). Table 3.1 summarizes the ordering strategies for black oil models with different numbers of phases. Note that the gas equations, whenever they are present, are always ordered last. This is because the gas component exists in both the oil and gas phases, so no ordering can produce the required triangular form when countercurrent flow is present.

Table 3.1: Ordering strategies for different black-oil models.

Model	Component ordering	Cell ordering		Primary pressure
		water	oil	
2-phase, oil-water	water/oil	Φ_w	*	p_w
2-phase, gas-water	water/gas	Φ_w	*	p_w
2-phase, oil-gas	oil/gas	*	Φ_o	p_o
3-phase	water/oil/gas	Φ_w	Φ_o	p_w

3.2.4 Remarks on implementation

In order to produce cell orderings that satisfy (3.2.7) and (3.2.8), it is not necessary to sort the potentials in decreasing order. Instead, consider the directed graph $G = (V, E)$ where the nodes V are the cells and the edges E are such that (i, j) is an edge whenever i and j are neighbors and $\Phi_i > \Phi_j$ or $\Phi_i = \Phi_j$ and $i > j$. Then a topological ordering of this graph (cf. [22]) will yield an ordering consistent with either (3.2.7) or (3.2.8), depending on which potential is used. The running time of this operation is $O(N)$, which is asymptotically faster than sorting ($O(N \log N)$).

We also remark that in most simulations, the flow directions do not change very often, so it may not be necessary to compute this ordering at every time step. For instance, we could compute the potential ordering only at the beginning of a time step. At each subsequent Newton iteration, we could simply verify the validity of the ordering, and only recompute it when the submatrix ceases to be triangular.

Chapter 4

Reduced Newton Method

In this chapter, we use the phase-based ordering introduced in section 3.2 to reformulate the mass-balance equations into a system of smaller size that involves pressure variables only. The Implicit Function Theorem [66] plays a central role in the formulation. We first describe the algorithm that arises when Newton's method is applied to the reduced system.

4.1 Algorithm description

For notational convenience, we rewrite (3.2.9) by splitting the equations into two blocks: the first block $F_s = 0$ contains all the water and oil equations, and the second block $F_g = 0$ contains all the gas equations. Similarly, we denote the vector of all saturation variables (S_{wi} and S_{oi}, $i = 1, \dots, N$) by S, and the vector of pressure variables by p. Then (3.2.9) becomes

$$\begin{cases} F_s(S,p) = 0 \\ F_g(S,p) = 0, \end{cases} \tag{4.1.1}$$

and the corresponding Jacobian J in (3.2.5) becomes

$$J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix}, \tag{4.1.2}$$

where

$$\begin{aligned} F_s &= [f_{w1}, \ldots, f_{wN}, f_{o1}, \ldots, f_{oN}]^T, \\ F_g &= [f_{g1}, \ldots, f_{gN}]^T, \\ S &= [S_{w1}, \ldots, S_{wN}, S_{o1}, \ldots, S_{oN}]^T, \\ p &= [p_{w1}, \ldots, p_{wN}]^T, \end{aligned}$$

and

$$J_{ss} = \partial F_s/\partial S, \quad J_{sp} = \partial F_s/\partial p, \quad J_{gs} = \partial F_g/\partial S, \quad J_{gp} = \partial F_g/\partial p.$$

It can be shown that J_{ss} is nonsingular as long as the monotonicity condition $dk_{rp}/dS_p \geq 0$ is valid for $p = o, w$ (see Appendix D for the proof). For $k_{rw} = k_{rw}(S_w)$ (which is usually obtained from experimental data), monotonicity is almost always satisfied, but the situation is less clear for $k_{ro} = k_{ro}(S_w, S_g)$, since the latter is usually obtained by interpolating data from oil-water and oil-gas experiments. Certain methods of interpolation, such as Stone I and Stone II [6], yield monotonic k_{ro} under mild conditions (see Appendix D), but this is not always the case for other methods (e.g., the segregation model [37]). In this work it is assumed that k_{ro} is a monotonically increasing function of S_o when S_w is fixed, which would ensure the nonsingularity of J_{ss}.

Consequently, since $F_s(S, p)$ has a triangular structure with respect to saturation, one can solve for S one unknown at a time if p is given. In addition, the implicit function theorem guarantees that if $F_s(S_0, p_0) = 0$ and $\partial F_s/\partial S$ is nonsingular at (S_0, p_0), then there exists a neighborhood U of p_0 and a unique differentiable function $S = S(p)$ such that $S(p_0) = S_0$ and $F_s(S(p), p) = 0$ for all $p \in U$. In other words, we can use F_s as a constraint to define saturation as a function of pressure, and substitute it into the remaining equations F_g. Thus, we obtain

$$F_g(S(p), p) = 0, \tag{4.1.3}$$

which we need to solve for the pressure p. If we use Newton's method to solve (4.1.3), the Jacobian becomes

$$J_{\text{reduced}} = \frac{\partial F_g}{\partial S}\frac{\partial S}{\partial p} + \frac{\partial F_g}{\partial p} \tag{4.1.4}$$

$$= J_{gs}\frac{\partial S}{\partial p} + J_{gp}. \tag{4.1.5}$$

Now $\partial S/\partial p$ is given by the implicit function theorem: $F_s(S(p), p) \equiv 0$ implies

$$\frac{\partial F_s}{\partial S}\frac{\partial S}{\partial p} + \frac{\partial F_s}{\partial p} = 0, \tag{4.1.6}$$

which we can write as

$$J_{ss}\frac{\partial S}{\partial p} + J_{sp} = 0. \tag{4.1.7}$$

Thus, the reduced Jacobian matrix is

$$J_{\text{reduced}} = J_{gp} - J_{gs}J_{ss}^{-1}J_{sp}, \tag{4.1.8}$$

which is precisely the Schur complement of (4.1.2) with respect to pressure. Figure 4.1 summarizes the algorithm used to solve the reduced system. Notice that the only difference between the algorithm in Figure 4.1 and Newton's method applied to the full problem is the way we compute S^{k+1}. In the full method, we set $S^{k+1} = S^k + \delta S^k$; in the reduced method, S^{k+1} is updated nonlinearly by solving the constraint equations $F(S^{k+1}, p^{k+1}) = 0$, in which the special triangular structure of J_{ss} is exploited. Also note that since this is just the usual Newton method applied to a reduced problem, *convergence is locally quadratic.*

Sequential updating of the saturations

The algorithm in Figure 4.1 requires the solution of $F_s(S^{k+1}, p_w^{k+1}) = 0$ for S^{k+1} at every step. Using the potential ordering presented in Section 3.2, we can triangularize the constraint equations to obtain the system (3.2.9). Thus, given the pressure values $p_1, \ldots, p_N$, we first solve $f_{w1} = 0$ for S_{w1}. Then, using this S_{w1} we can now solve $f_{w2} = 0$ for S_{w2}, and so on until we obtain all saturation values. Thus, solving $F_s(S^{k+1}, p_w^{k+1}) = 0$ reduces to solving $(n_p - 1)N$ nonlinear scalar equations one at

```
while $\left|F_g(S(p_w^k), p_w^k)\right| > tol$ , do
    Form the full Jacobian $J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix}$ , evaluated at $(S(p_w^k), p_w^k)$ ;
    Solve $(J_{gp} - J_{gs} J_{ss}^{-1} J_{sp}) \delta p^k = -r^k$ ;
    Compute $p_w^{k+1} = p_w^k + \delta p^k$ ;
    Update $S^{k+1} = S(p_w^{k+1})$ nonlinearly by solving $F_s(S^{k+1}, p_w^{k+1}) = 0$,
        one variable at a time in potential ordering ;
    $k := k + 1$
end
```

Figure 4.1: Algorithm for solving the reduced system (4.1.3).

a time (where n_p is the number of fluid phases). A wide variety of reliable univariate solvers are available to deal with the nonlinear single-cell problems. One such choice is the van Wijngaarden-Dekker-Brent Method [14], which combines bisection with inverse quadratic interpolation to obtain superlinear convergence. This is a derivative-free algorithm, which means only function values are required, although an initial guess based on the solution of the ordinary Newton step can be used to accelerate convergence. In a reasonably efficient implementation, each function evaluation should only require a few floating-point operations. As shown in section 4.3, the extra cost of the single-cell nonlinear solves is usually offset by a reduction in the number of global Newton steps. The nonlinear updates can be performed quite efficiently if more sophisticated zero-finders are used.

Solving the Schur complement system

There are two ways to solve the Schur complement system

$$J_{\text{reduced}} \delta p = -r. \tag{4.1.9}$$

The first way is to notice that one can solve the equivalent system

$$\begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix} \begin{bmatrix} \delta S \\ \delta p \end{bmatrix} = \begin{bmatrix} 0 \\ -r \end{bmatrix}. \tag{4.1.10}$$

Krylov subspace methods (such as GMRES) can be used, and effective preconditioners (such as the Constrained Pressure Residual method [81]) are available. A second way is to apply the Krylov method directly to the Schur complement system. In this approach, matrix-vector multiplication by J_{reduced} would have the same cost as multiplication by the full matrix, because J_{ss} is lower triangular, so that multiplication by J_{ss}^{-1} is simply a forward substitution. In terms of preconditioning, one can either precondition J_{reduced} directly with ILU type methods, or use an *induced* preconditioner based on the full system by letting

$$M_{\text{reduced}}^{-1} = RM_{\text{full}}^{-1}R^T, \tag{4.1.11}$$

where M_{full}^{-1} is the preconditioner for the full system, and $R = \begin{bmatrix} 0 & I \end{bmatrix}$ is the restriction operator to the pressure variables. In other words, a preconditioning step for the reduced system $y = M_{\text{reduced}}^{-1}x$ consists of the following steps:

1. Pad the vector x with zeros to form $\hat{x} = \begin{pmatrix} 0_{(n_p-1)N} \\ x \end{pmatrix}$.

2. Compute $\hat{y} = M_{\text{full}}^{-1}\hat{x}$.

3. Let y be the portion of $\hat{y}$ corresponding to pressure variables, i.e., retain only the last N elements of $\hat{y}$.

One potential advantage of applying the Krylov method to the Schur complement system rather than the full system is that the resulting Krylov vectors are only of length N rather than length n_pN, where n_p is the number of fluid phases. This greatly reduces storage requirements and orthogonalization cost in methods such as GMRES, so that more Krylov steps can be taken before restarting.

In fact, the Schur complement reduction can be used even if the nonlinear constraint equations are not exactly satisfied. This could happen if the initial pressure guess is so poor that some of the residual constraints in the reduced Newton cannot be satisfied. In that case we would have

$$\begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix} \begin{bmatrix} \delta S \\ \delta p \end{bmatrix} = - \begin{bmatrix} r_s \\ r_g \end{bmatrix} \tag{4.1.12}$$

But this is equivalent to solving

$$J_{\text{reduced}}\delta p = -(r_g - J_{gs}J_{ss}^{-1}r_s) \tag{4.1.13}$$

which has the same form as (4.1.9). All these options are evaluated in chapter 5.

4.2 Convergence analysis

This section is devoted to the analysis of the reduced Newton method. For simplicity we assume we are dealing with the discrete version of the two-phase, 1D model problem outlined in section 2.2.1. Though simple, this model problem captures the nature of the nonlinearity of the reduced objective function for transport in porous media. A physical argument (supported by numerical evidence [82]) suggests that nonlinearity due to pressure is negligible unless highly compressible components (such as gas) are present in the system.

There are two basic mechanisms that guarantee the convergence of Newton's method. The first mechanism is contraction, i.e., when the Newton mapping

$$g : x \mapsto x - (f'(x))^{-1}f(x)$$

is contractive. Classical convergence theorems of this type include the Newton-Kanterovich and Newton-Mysovskikh theorems [29]. When the objective function f is a scalar function, the proofs simplify, and the following theorem can be established.

Theorem 4.1. *Let f be a C^2 function over some interval J, and let $I = (a, b) \subset J$ be an open interval such that $f'(x) \neq 0$ on I and*

$$x \in I \implies \frac{|f(x)f''(x)|}{|f'(x)|^2} < 1. \tag{4.2.1}$$

Let $x^* \in I$ be such that $f(x^*) = 0$, and let $L = \min\{|x^* - a|, |b - x^*|\}$. Then x^* is the unique root of f in I, and for any initial guess $x_0 \in (x^* - L, x^* + L)$, the Newton *iteration*

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

converges quadratically to x^**.*

The above theorem, while establishing quadratic convergence, is inherently a local result. This is because the quantity $|f(x)f''(x)/f(x)^2|$ is typically small only in the vicinity of a root. Since our goal is to prove global convergence over a very wide parameter space of relative permeability functional forms and initial guesses, it is very difficult to ensure that criterion (4.2.1) is satisfied in every case. Thus, we must exploit the other mechanism for convergence, namely, the convexity of the objective function.

4.2.1 Convex functions and Newton's method

Recall that a function $f : [a, b] \to \mathbb{R}$ is *convex* if, for all $x, y \in [a, b]$ and $0 \leq t \leq 1$, we have

$$f((1-t)x + ty) \leq (1-t)f(x) + tf(y).$$

Convex functions enjoy many nice properties (such as continuity everywhere and differentiability almost everywhere [59]), but for our purposes we mainly consider C^2 functions. The following properties are used repeatedly in our analysis.

Lemma 4.2 (Properties of convex functions [59, §3.4])**.** *Let* $f : [a, b] \to \mathbb{R}$ *be a* C^2 *function. Then the following are equivalent:*

1. f *is convex on* $[a, b]$*;*

2. $f'(x)(y - x) \leq f(y) - f(x)$ *for all* $x, y \in [a, b]$*;*

3. $f''(x) \geq 0$ *for all* $x \in [a, b]$*.*

Theorem 4.3 (Monotone convergence of Newton's method)**.** *Let* $f : \mathbb{R} \to \mathbb{R}$ *be a* C^2 *function such that* $f'(x) > 0$ *everywhere, and let* x^* *be such that* $f(x^*) = 0$*. Suppose there is a semi-infinite interval* $I = [c, \infty)$ *such that* $x^* \in I$ *and* $f''(x) \geq 0$ *for all*

$x \in I$. Then Newton's method converges to x^* for an arbitrary initial guess x_0. In addition, if $f(x_0) \geq 0$, then the Newton iterates converge monotonically, i.e.

$$x_0 \geq x_1 \geq \cdots \geq x_k \geq \cdots \geq x^*.$$

Proof. First, assume that $f(x_k) \geq 0$ for some $k \geq 0$. Then f is convex on the interval $[x^*, x_k]$, so we have

$$f'(x_k)(x_k - x) \leq f(x_k) - f(x^*) = f(x_k).$$

Since $f'(x_k) > 0$, rearranging gives

$$x^* \geq x_k - \frac{f(x_k)}{f'(x_k)} = x_{k+1},$$

which, together with the fact that $f(x_k) \geq 0$, implies $x^* \leq x_{k+1} \leq x_k$. So $f(x_{k+1}) \geq 0$ and f is convex on $[x^*, x_{k+1}]$. Induction now shows that

$$x_k \geq x_{k+1} \geq \cdots \geq \cdots \geq x^*,$$

which means $\{x_k\}_{k=1}^{\infty}$ is a decreasing sequence bounded below by x^*; thus, the sequence converges to a limit $\tilde{x}$. Since the Newton mapping $g : x \mapsto x - (f'(x))^{-1} f(x)$ is continuous, we must have

$$\tilde{x} = \lim g(x_k) = g(\tilde{x}),$$

so that

$$\tilde{x} = \tilde{x} - (f'(\tilde{x}))^{-1} f(\tilde{x}),$$

which implies $f(\tilde{x}) = 0$. Hence $\tilde{x} = x^*$, and Newton's method converges to x^*. The second statement of the theorem now follows, since it is just the special case $k = 0$.

Now assume, on the contrary, that there is no k such that $f(x_k) \geq 0$. In other words, we have $f(x_k) < 0$ for every $k \geq 0$. But this implies:

1. $x_k < x^*$ for all k, since f is monotonically increasing, and

2. $x_{k+1} > x_k$ for all k, since $f(x_k) < 0$.

This means $\{x_k\}_{k=1}^{\infty}$ is an increasing sequence bounded above by x^*, so it converges to some limit, which must then be equal to x^* by the continuity of the Newton mapping. So in both cases, Newton's method converges to the root x^*, as required. □

To exploit this useful connection between convex functions and Newton convergence, we make the following assumptions on the relative permeability functions.

Assumption 4. We assume that the following properties hold for all saturations $0 \le S_w \le 1$:

1. $\lambda_T(S_w) = k_w(S_w)/\mu_w + k_o(S_w)/\mu_o > 0$ (Uniform ellipticity),

2. $k_w'(S_w) \ge 0$, $k_o'(S_w) \le 0$ (Phase mobilities increasing with phase saturations),

3. $k_w''(S_w) \ge 0$, $k_o''(S_w) \ge 0$ (Convex relperms).

Uniform ellipticity is an essential assumption that is required for the well-posedness of the elliptic subproblem. The requirements on k_{rw}' and k_{ro}' are the same as those in chapter 2, which, as indicated previously, are physically realistic. Convexity of the relative permeabilities is the only additional assumption, and most commonly used relative permeability functions, such as those due to Honarpour *et al.* [42], satisfy this requirement.

4.2.2 The cocurrent case: large Δt

Recall that in the cocurrent case, the upstream direction is the same for both phases, i.e., $\lambda_{p,i+1/2} = \lambda_p(S_i)$ for $p = o, w$. For analysis purposes, we perform a linear change of variables by defining $\pi_i = (p_i - p_{i+1})/\Delta x$, the pressure gradient at the cell interface $i + 1/2$. Then the mass balance equations become

$$\begin{aligned} F_{wi} &= V_i S_i - K_{i-1}\lambda_w(S_{i-1})\pi_{i-1} + K_i\lambda_w(S_i)\pi_i - q_{wi}, \\ F_{oi} &= -V_i S_i - K_{i-1}\lambda_o(S_{i-1})\pi_{i-1} + K_i\lambda_o(S_i)\pi_i - q_{oi}, \end{aligned} \tag{4.2.2}$$

where $V_i = \phi_i \Delta x/\Delta t$ and K_i is the (absolute) permeability between blocks i and $i+1$. We note that applying Newton's method to this modified system will yield pressure profiles that are identical to those obtained from applying Newton's method to the original system, since all we did is a linear change of independent variables.

Now consider applying reduced Newton to (4.2.2), i.e., we use the water phase equations as the constraints required to define the implicit functions

$$S_i = S_i(\pi_1, \dots, \pi_i).$$

Then we can rewrite (4.2.2) as

$$\begin{aligned} F_{wi}(\pi_1, \dots, \pi_i) &= V_i S_i(\pi_1, \dots, \pi_i) + K_i \lambda_w(S_i(\pi_1, \dots, \pi_i))\pi_i - f_{wi}(\pi_1, \dots, \pi_{i-1}) \equiv 0, \\ F_{oi}(\pi_1, \dots, \pi_i) &= -V_i S_i(\pi_1, \dots, \pi_i) + K_i \lambda_o(S_i(\pi_1, \dots, \pi_i))\pi_i - f_{oi}(\pi_1, \dots, \pi_{i-1}), \end{aligned} \tag{4.2.3}$$

where f_{wi} and f_{oi} are influxes from the upwind cell, which do not depend on the pressure gradient π_i. Thus, our approach for proving convergence is as follows: we show that for fixed $\pi_1, \dots, \pi_{i-1}$, the objective function F_{oi} is strictly increasing and convex with respect to π_i over a semi-infinite interval containing the root π_i^*. Thus, Newton's method converges for any starting point within this interval. Then an induction argument, together with the continuous dependence of F_{oi} on the influx $f_{oi}(\pi_1, \dots, \pi_{i-1})$, will guarantee global convergence of Newton's method for the whole system.

Remark. Without loss of generality, we can restrict our attention to how reduced Newton behaves inside the positive orthant $\{\pi_i > 0, i = 1, \dots, N\}$. Let π_i^* denote the solution of the i-th cell problem (so that $F_{oi}(\pi_1^*, \dots, \pi_i^*) = 0$). Since flow is cocurrent and the total velocity is positive, each π_i^* must be positive. Moreover, because of uniform ellipticity, we have the lower bound

$$\pi_i^* \geq \frac{q}{K_i(\lambda_T)_{\max}} = \frac{q \min\{\mu_o, \mu_w\}}{K_i}, \tag{4.2.4}$$

where the last equality holds by convexity.

We are now ready to show that reduced Newton converges when Δt is large, provided we make a few additional assumptions that are satisfied by quadratic relative permeabilities. In the next section, we derive a modified reduced Newton iteration that is provably convergent for all Δt without the need of these additional assumptions.

Proposition 4.4. *Assume k_w and k_o are both uniformly convex, i.e., there exist positive constants c_w and c_o such that $k_w'' \geq c_w$ and $k_o'' \geq c_o$ for all $S \in [0, 1]$. Let $k_w'(0) = 0$. Then there exists $S_c > 0$ such that $\lambda_w' + \lambda_o' \leq 0$ for all $0 \leq S_w \leq S_c$.*

Proof. Since $k_o'(S_w = 1) \leq 0$ and $k_o''(S_w) \geq c_o > 0$, we must have $k_o'(S_w = 0) \leq -c_o$, so that $\lambda_w' + \lambda_o' \leq -c_o/\mu_o < 0$. Thus, by continuity, there exists a non-trivial neighborhood around zero, say $0 \leq S_w \leq S_c$, such that $\lambda_w' + \lambda_o'$ takes on negative values. □

Lemma 4.5 (Monotonicity and convexity with respect to π_i)**.** *Assume the hypotheses of Proposition 4.4. Let $\pi_j > 0$ for all j. Then $\partial F_{oi}/\partial \pi_i > 0$, and there exists $\gamma_0 > 0$ such that $\partial^2 F_{oi}/\partial \pi_i^2 \geq 0$ whenever $V_i/K_i\pi_i \leq \gamma_0$.*

Proof. The water phase constraint yields

$$\frac{\partial S_i}{\partial \pi_i} = -\frac{K_i \lambda_w}{V_i + K_i \lambda_w' \pi_i},$$

implying that

$$\frac{\partial F_{oi}}{\partial \pi_i} = K_i \lambda_o + \frac{K_i \lambda_w (V_i - K_i \lambda_o' \pi_i)}{V_i + K_i \lambda_w' \pi_i},$$

which is positive for $\pi_i > 0$ if the fluid properties in Assumption 4 hold. Similarly, the second derivative is given by

$$\begin{aligned}\frac{\partial^2 F_{oi}}{\partial \pi_i^2} = -\frac{K_i \lambda_w}{(V_i + K_i \lambda_w' \pi_i)^2} \Bigg\{ & 2 V_i K_i (\lambda_w' + \lambda_o') \\ & - \frac{K_i \lambda_w}{V_i + K_i \lambda_w' \pi_i} \left[K_i^2 \pi_i^2 (\lambda_o'' \lambda_w' - \lambda_w'' \lambda_o') + V_i K_i \pi_i (\lambda_w'' + \lambda_o'') \right] \Bigg\}.\end{aligned}$$

The terms inside the square brackets are non-negative by Assumption 4. Thus, if

$\lambda_w' + \lambda_o' \le 0$, then $\partial^2 F_{oi}/\partial \pi_i^2 \ge 0$ automatically. If $\lambda_w' + \lambda_o' > 0$, then we need

$$2V_iK_i(\lambda_w' + \lambda_o') \le \frac{K_i\lambda_w}{V_i + K_i\lambda_w'\pi_i}\big[K_i^2\pi_i^2(\lambda_o''\lambda_w' - \lambda_w''\lambda_o') + V_iK_i\pi_i(\lambda_w'' + \lambda_o'')\big].$$

Cross-multiplying and setting $\gamma = V_i/K_i\pi_i$ gives

$$A\gamma^2 + B\gamma + C \le 0 \tag{4.2.5}$$

with

$$\begin{aligned} A &= 2(\lambda_w' + \lambda_o'),\\ B &= 2\lambda_w'(\lambda_w' + \lambda_o') - \lambda_w(\lambda_w'' + \lambda_o''),\\ C &= -\lambda_w(\lambda_o''\lambda_w' - \lambda_w''\lambda_o'). \end{aligned}$$

Since $A > 0$ and $C < 0$, we deduce that (4.2.5) is satisfied iff

$$\gamma \le \frac{-B + \sqrt{B^2 - 4AC}}{2A} = \frac{-2C}{B + \sqrt{B^2 - 4AC}}.$$

Thus, $\partial^2 F_{oi}/\partial \pi_i^2 \ge 0$ if $\gamma \le \gamma_0$, where

$$\gamma_0 = \min_{S_c \le S \le 1} \frac{-2C}{B + \sqrt{B^2 - 4AC}}. \tag{4.2.6}$$

We exclude the interval $[0, S_c)$ from the minimization because $\lambda_w' + \lambda_o' < 0$ there. This implies $\gamma_0 > 0$, since

$$-C \ge \lambda_w(S_c)\lambda_w'(S_c)\lambda_{o,\min}'' \ge \frac{c_w^3 c_o}{2\mu_w^2\mu_o} > 0$$

and the denominator is bounded. □

Lemma 4.5 implies that if $V_i \le \theta\gamma_0 q \min\{\mu_o, \mu_w\}$ with $\theta < 1$, then F_{oi} is convex in the interval $\theta q \min\{\mu_o, \mu_w\} \le \pi_i < \infty$, which contains π_i^*. Hence, by Theorem 4.3, the sequence of Newton iterates $\{\pi_i^{(k)}\}$ converges monotonically to π_i^* provided the influx is constant. In particular, the first cell converges if Δt is *large enough*. Global convergence follows by induction and continuity.

4.2.3 The general cocurrent case

The rather weak result on convergence in the previous section is due to our inability to ascertain convexity of the objective function except under fairly limited circumstances. Figure 4.2 plots the objective function $F_{oi}(\pi_i)$ for different time-step sizes. We see that for large Δt, the objective function is indeed convex over a semi-infinite interval containing the root, but this is not always the case for smaller Δt, especially for unfavorable mobility ratios. In practice, our numerical results show that convergence still occurs, but this is due to contraction rather than convexity. In order to ensure global convergence based on a convexity argument, we need to make a small modification to the reduced Newton algorithm. The following lemma is the key observation.

Lemma 4.6. *Let $(\pi_1, \dots, \pi_N) > 0$ be given. Suppose we define the implicit functions $S_i^{(1)}(\pi_1, \dots, \pi_i)$ and $S_i^{(2)}(\pi_1, \dots, \pi_i)$ via the constraints*

$$\begin{aligned} F_{wi}(\pi_1, \dots, \pi_i) &= V_i S_i^{(1)}(\pi_1, \dots, \pi_i) + K_i \lambda_w(S_i^{(1)}(\pi_1, \dots, \pi_i))\pi_i - f_{wi}(\pi_1, \dots, \pi_{i-1}) \equiv 0, \\ F_{oi}(\pi_1, \dots, \pi_i) &= -V_i S_i^{(2)}(\pi_1, \dots, \pi_i) + K_i \lambda_o(S_i^{(2)}(\pi_1, \dots, \pi_i))\pi_i - f_{oi}(\pi_1, \dots, \pi_{i-1}) \equiv 0, \end{aligned} \tag{4.2.7}$$

respectively. Now consider the reduced functions

$$\begin{aligned} \bar{F}_{oi}(\pi_1, \dots, \pi_i) &= -V_i S_i^{(1)}(\pi_1, \dots, \pi_i) + K_i \lambda_o(S_i^{(1)}(\pi_1, \dots, \pi_i))\pi_i - f_{oi}(\pi_1, \dots, \pi_{i-1}), \\ \bar{F}_{wi}(\pi_1, \dots, \pi_i) &= V_i S_i^{(2)}(\pi_1, \dots, \pi_i) + K_i \lambda_w(S_i^{(2)}(\pi_1, \dots, \pi_i))\pi_i - f_{wi}(\pi_1, \dots, \pi_{i-1}). \end{aligned} \tag{4.2.8}$$

Then both $\bar{F}_{oi}$ and $\bar{F}_{wi}$ are increasing with respect to π_i, and at least one of $\bar{F}_{oi}$ and $\bar{F}_{wi}$ must be a convex function over a semi-infinite interval containing the root π_i^*.

Proof. We have shown in Lemma 4.5 that

$$\frac{\partial S_i^{(1)}}{\partial \pi_i} = -\frac{K_i \lambda_w}{V_i + K_i \lambda_w' \pi_i},$$

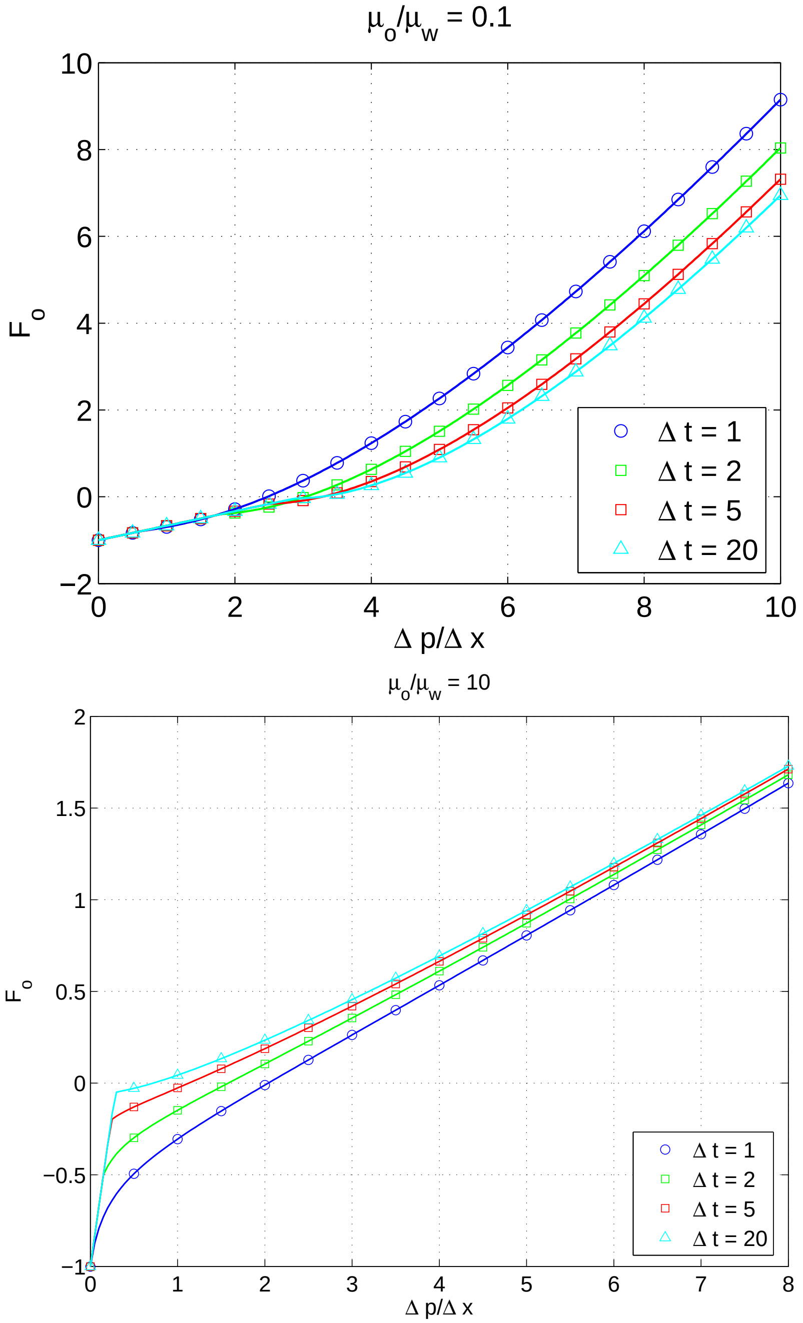


Figure 4.2: Reduced Newton residual functions for various Δt. Top: favorable mobility ratio ($\mu_o/\mu_w = 0.1$). Bottom: unfavorable mobility ratio ($\mu_o/\mu_w = 10$).

and that the first and second derivatives of $\bar{F}_{oi}$ are

$$\frac{\partial \bar{F}_{oi}}{\partial \pi_i} = K_i \lambda_o + \frac{K_i \lambda_w (V_i - K_i \lambda_o' \pi_i)}{V_i + K_i \lambda_w' \pi_i}$$

and

$$\begin{aligned} \frac{\partial^2 \bar{F}_{oi}}{\partial \pi_i^2} = -\frac{K_i \lambda_w}{(V_i + K_i \lambda_w' \pi_i)^2} \Big\{ & 2 V_i K_i (\lambda_w' + \lambda_o') \\ & - \frac{K_i \lambda_w}{V_i + K_i \lambda_w' \pi_i} \big[K_i^2 \pi_i^2 (\lambda_o'' \lambda_w' - \lambda_w'' \lambda_o') + V_i K_i \pi_i (\lambda_w'' + \lambda_o'') \big] \Big\}, \end{aligned}$$

where the λ_p and their derivatives are evaluated at $S_i^{(1)}(\pi_1, \ldots, \pi_i)$. A similar calculation shows that

$$\begin{aligned} \frac{\partial S_i^{(2)}}{\partial \pi_i} &= \frac{K_i \lambda_o}{V_i - K_i \lambda_o' \pi_i}, \\ \frac{\partial \bar{F}_{wi}}{\partial \pi_i} &= K_i \lambda_w + \frac{K_i \lambda_o (V_i + K_i \lambda_w' \pi_i)}{V_i - K_i \lambda_o' \pi_i} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial^2 \bar{F}_{wi}}{\partial \pi_i^2} = \frac{K_i \lambda_o}{(V_i - K_i \lambda_o' \pi_i)^2} \Big\{ & 2 V_i K_i (\lambda_w' + \lambda_o') \\ & + \frac{K_i \lambda_o}{V_i - K_i \lambda_o' \pi_i} \big[K_i^2 \pi_i^2 (\lambda_o'' \lambda_w' - \lambda_w'' \lambda_o') + V_i K_i \pi_i (\lambda_w'' + \lambda_o'') \big] \Big\}, \end{aligned}$$

where the λ_p and their derivatives are now evaluated at $S_i^{(2)}(\pi_1, \ldots, \pi_i)$. By definition, at the solution π_i^* we must have $S_i^{(1)}(\pi_1, \ldots, \pi_i^*) = S_i^{(2)}(\pi_1, \ldots, \pi_i^*) =: S_i^*$. Moreover, we must have $S_i^{(1)} \leq S_i^* \leq S_i^{(2)}$ over the interval $[\pi_i^*, \infty)$ because $\partial S_i^{(1)} / \partial \pi_i \leq 0$ and $\partial S_i^{(2)} / \partial \pi_i \geq 0$. We now consider two cases:

1. $\lambda_w'(S_i^*) + \lambda_o'(S_i^*) \geq 0$. Then since $S_i^{(2)} \geq S_i^*$ for $\pi_i \geq \pi_i^*$, the convexity of λ_w and λ_o implies $\lambda_w'(S_i^{(2)}) + \lambda_o'(S_i^{(2)}) \geq 0$. Hence $\partial \bar{F}_{wi} / \partial \pi_i \geq 0$ for all $\pi_i \geq \pi_i^*$.

2. $\lambda_w'(S_i^*) + \lambda_o'(S_i^*) \leq 0$. Then since $S_i^{(1)} \leq S_i^*$ for $\pi_i \geq \pi_i^*$, the convexity of λ_w and λ_o implies $\lambda_w'(S_i^{(1)}) + \lambda_o'(S_i^{(1)}) \leq 0$. Hence $\partial \bar{F}_{oi} / \partial \pi_i \geq 0$ for all $\pi_i \geq \pi_i^*$.

Thus, at least one of the two reduced functions is convex over a semi-infinite interval containing π_i^*, as required. □

The above lemma tells us that if we knew ahead of time the slope of the total mobility curve at the solution, we could always pick the correct reduced function (or equivalently, the correct constraint) for each cell in order to achieve global convergence. Unfortunately, this information is usually not available. However, if we switch constraints when non-convexity is detected, then we can be certain that the new reduced function must be convex, so convergence is now guaranteed. The modified algorithm is shown in Figure 4.3. The convexity test in line 9 is motivated by Theorem 4.3. Assume all cells upstream of i have converged. If the current residual function is convex and $F_{gi}(\pi_i^k) > 0$, then we should have $F_{gi}(\pi_i^{k+1}) > 0$ as well. Thus, if the latter condition is violated, non-convexity is detected, so we should switch constraints and work with the other residual function, which must be convex. In practice, we may not want to swap constraints every time the residual becomes negative for the following reasons:

- The upstream cells may not have converged;
- When the nonlinear iterate is close to the solution (but has not yet converged), the residual can have the wrong sign even when convex objective functions are used. This is because the linear and nonlinear equations that define the Newton steps are themselves solved inexactly by inner iterations;
- Frequent constraint switches can lead to a deterioration in global convergence.

As a result, we should switch constraints only when the overshoot is severe enough that we are certain no progress has been made. The parameter $0 < \theta < 1$ in line 9 achieves this purpose: if the new residual changes sign but has a significantly smaller magnitude, we accept the current constraint and continue; on the other hand, large overshoots cause the constraint to switch. It is fairly easy to convince oneself that the modified algorithm converges for all initial guesses inside the positive orthant $\{\pi_i > 0, i = 1, \ldots, N\}$.

```
Initialize constraint set $s := \{F_{w1}, \ldots, F_{wN}\}$ and its complement $g := s'$
while $\left|F_g(S^k), p^k)\right| > tol$ , do
    Form the full Jacobian $J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{gs} & J_{gp} \end{bmatrix}$ , evaluated at $(S^k, p^k)$ ;
    Solve $(J_{gp} - J_{gs} J_{ss}^{-1} J_{sp}) \delta p^k = -r^k$ ;
    Compute $p^{k+1} = p^k + \delta p^k$ ;
    Update $S^{k+1}$ nonlinearly by solving $F_s(S^{k+1}, p^{k+1}) = 0$,
        one variable at a time in potential ordering;
    for $i = 1, \ldots, n$ , do
        if $F_{gi}(S^{k+1}, p^{k+1}) < -\theta F_{gi}(S^k, p^k)$ , then
            $s := s \cup \{F_{gi}\} \backslash \{F_{si}\}$   (Swap constraints)
            $g := g \cup \{F_{si}\} \backslash \{F_{gi}\}$
        end if
    end
    $k := k + 1$
end
```

Figure 4.3: Modified reduced Newton algorithm.

4.2.4 The countercurrent flow case

When gravity effects are included, countercurrent flow may be present in some parts of the domain. In a cell experiencing countercurrent flow, the mass balance equations take the form

$$\begin{aligned} F_{wi} &= V_i S_i - K_{i-1} \lambda_w(S_{i-1}) \pi_{i-1} + K_i \lambda_w(S_i) \pi_i - q_{wi}, \\ F_{oi} &= -V_i S_i - K_{i-1} \lambda_o(S_i)(\pi_{i-1} - \Delta\rho g \Delta z) + K_i \lambda_o(S_{i+1})(\pi_i - \Delta\rho g \Delta z) - q_{oi}. \end{aligned} \tag{4.2.9}$$

Here $\pi_i = (\Phi_{w,i} - \Phi_{w,i+1})/\Delta x$ denotes the gradient for the water potential. The presence of countercurrent flow introduces several complications in our attempt to analyze the convergence behavior of the reduced Newton algorithm:

1. Convergence can no longer be analyzed by considering a sequence of independent single-cell problems. Since the objective functions F_{oi} now contain downstream dependencies, all the cells within the countercurrent flow region are fully coupled.

2. The flow direction of the oil phase, which depends on $\pi_i^* - \Delta\rho g \Delta z$, is generally not known until the problem has converged.

3. The objective function becomes non-differentiable when the upstream direction changes.

Our experiments show that when significant countercurrent flow is present, it is possible that reduced Newton no longer converges to the solution for every initial guess, especially when a large time step is taken. It is then natural to try to identify conditions for which the reduced Newton procedure converges.

A domain of dependence argument

To derive a criterion that would ensure convergence, we turn to a heuristic argument based on the domain of dependence. In the theory of numerical methods for hyperbolic PDEs, the Courant-Friedrichs-Lewy (CFL) condition states that if a numerical method is stable, then its numerical domain of dependence must be at least as large as the domain of dependence of the underlying PDE (cf. [49]). In this context, the superior convergence behavior of reduced Newton for cocurrent flow can be explained as follows: the implicit function S_i, defined by the water phase constraint

$$F_{wi} = V_i(S_i - S_i^0) - K_{i-1}\lambda_w(S_{i-1})\pi_{i-1} + K_i\lambda_w(S_i)\pi_i - q_{wi} \equiv 0,$$

is actually a function of the arguments

$$S_i = S_i(\pi_1, \ldots, \pi_i; S_1^0, \ldots, S_i^0),$$

where $\{S_i^0\}$ denotes the *initial saturation profile*, i.e., the saturation profile at the beginning of the time step. Thus, the objective function F_{oi} actually depends implicitly on the old saturation values $S_1^0, \ldots, S_i^0$ as well as the pressure gradients $\pi_1, \ldots, \pi_i$. Since the characteristics of the PDE only travel from left to right in the cocurrent case, the "domain of dependence" of reduced Newton contains the domain of dependence of the PDE for any Δt. As a result, one can expect a fairly stable method for a wide range of initial guesses. On the other hand, in the countercurrent flow case, the objective function F_{oi} takes the form

$$\begin{aligned} F_{oi} &= -V_i S_i - K_{i-1}\lambda_o(S_i)(\pi_{i-1} - \Delta\rho g \Delta z) + K_i \lambda_o(S_{i+1})(\pi_i - \Delta\rho g\Delta z) - q_{oi} \\ &= F_{oi}(S_i(\cdots), S_{i+1}(\cdots), \pi_{i-1}, \pi_i) \\ &= F_{oi}(\pi_1, \ldots, \pi_{i+1}; S_1^0, \ldots, S_{i+1}^0). \end{aligned}$$

Thus, if Δt is so large that the waves traveling to the left (i.e., countercurrent to the main flow direction) can cross more than one cell boundary, then the domain of dependence of reduced Newton will fail to cover the physical domain of dependence. In such cases, one cannot generally expect global convergence of the reduced Newton iterations. Since the fastest backward-moving wave travels at the speed of $v_{\min} = \min_{S\in[0,1]} q_T f_w'$, where

$$f_w = \frac{\lambda_w}{\lambda_T}\left[1 + \frac{K_i\lambda_o}{q_T}\Delta\rho g\Delta z\right], \tag{4.2.10}$$

we can expect reduced Newton to converge whenever

$$-\Delta t q_T f'_{w,\min} \leq \phi_i \Delta x. \tag{4.2.11}$$

Thus, if $f_w' \geq 0$ everywhere (i.e., we have cocurrent flow), we expect reduced Newton to converge for any Δt. If countercurrent flow is present, then there is a range of S over which $f_w' < 0$, in which case we would have the time-step restriction

$$\Delta t \leq \frac{\phi_i \Delta x}{q_T \left|f'_{w,\min}\right|}, \tag{4.2.12}$$

which is effectively a CFL limit for backward-traveling waves.

A monotonicity argument

We have shown in Lemma 4.5 that in the cocurrent case, the objective function is monotonically increasing ($\partial F_{oi}/\partial \pi_i > 0$). Monotonicity is an important property if global convergence to a unique solution is to be expected: non-monotonic functions necessarily have local minima or maxima, which cause breakdown in Newton's method. Thus, a reasonable criterion for ensuring convergence is one that guarantees monotonicity of the objective function. We can mimic the proof of Lemma 4.5 and compute the partial derivative $\partial \hat{F}_{oi}/\partial \pi_i$, where

$$\hat{F}_{oi} = -V_i S_i + K_i \lambda_o(S_i)(\pi_i - \Delta\rho g \Delta z) - f_{oi}(\pi_1, \dots, \pi_{i-1}).$$

In other words, we perform the analysis as though the upstream direction is to the left. Even though this upstream direction may be incorrect, the analysis is still valuable for the following reason: since the correct upstream direction is generally unknown before the solution has converged, a robust algorithm should still be able to make some progress even when the upstream direction is wrong. The algorithm should produce an answer that would cause a switch in the upstream direction in the next iteration, but it should not overshoot by so much as to cause the overall algorithm to fail. These desirable properties are only possible when $\hat{F}_{oi}$ is monotonic, so our analysis can still provide a useful criterion for convergence.

We have

$$\begin{aligned}\frac{\partial \hat{F}_{oi}}{\partial \pi_i} &= -V_i \frac{\partial S_i}{\partial \pi_i} + K_i \lambda_o + K_i \lambda_o'(\pi_i - \Delta\rho g \Delta z) \frac{\partial S_i}{\partial \pi_i}, \\ &= \frac{K_i}{V_i + K_i \lambda_w' \pi_i} \Big[\lambda_T V_i + K_i \lambda_o \lambda_w' \pi_i - K_i \lambda_w \lambda_o' (\pi_i - \Delta\rho g \Delta z) \Big].\end{aligned}$$

Using the relations

$$\begin{aligned}\pi_i &= \frac{1}{\lambda_T} \big[q_T/K_i + \lambda_o \Delta\rho g \Delta z \big] \\ \pi_i - \Delta\rho g \Delta z &= \frac{1}{\lambda_T} \big[q_T/K_i - \lambda_w \Delta\rho g \Delta z \big],\end{aligned}$$

we can rewrite $\partial \hat{F}_{oi}/\partial \pi_i$ as

$$\frac{\partial \hat{F}_{oi}}{\partial \pi_i} = \frac{K_i \lambda_T}{V_i + K_i \lambda_w' \pi_i} \Big[V_i + q_T f_w' \Big],$$

where $f_w(S)$ is defined in (4.2.10). Thus, the objective function $\hat{F}_{oi}$ is monotonically increasing whenever

$$V_i = \frac{\phi_i \Delta x}{\Delta t} \geq -q_T f_w',$$

which is exactly the same as (4.2.11). As it is shown in Example 4.3.1, criterion (4.2.11) is usually enough for reduced Newton to converge. For problems of practical interest, the backward CFL number is usually much smaller than the forward CFL number, so reduced Newton can generally converge with much larger time steps than standard Newton even in the countercurrent flow setting. In the next section, we show a variety of examples that demonstrate the effectiveness of the reduced Newton algorithm.

4.3 Numerical examples

To test the efficiency of the potential-based reduced Newton algorithm, we implement it inside the General Purpose Research Simulator (GPRS) developed by Cao [16]. GPRS is used by Stanford University's SUPRI-B and SUPRI-HW research groups, as well as other research groups and companies for their in-house research. By implementing our algorithm in GPRS we can guarantee that all the property calculations and convergence checks are identical for both the standard and reduced Newton methods. We can also ensure our reference point is indeed the basic Newton method, rather than a version adorned with various heuristics. Consequently, all our comparisons between the standard and reduced Newton methods are generated by GPRS.

4.3.1 1D example with gravity

To demonstrate that the potential-based reduced Newton algorithm does indeed work in the presence of countercurrent flow, we first test it on a simple pseudo-1D example. The reservoir is discretized using 10 × 1 × 100 cells in the x, y and z directions respectively, with $D_x = 10$ ft, $D_y = 50$ ft and $D_z = 4$ ft. A uniform porosity ($\phi = 0.3$) and permeability ($k_x = k_y = k_z = 758$ md) are used. Water is injected across the top layer at a rate of 213.6 bbl/day (0.002 pore volumes per day) and a production well is completed across the bottom layer and operates at a BHP (bottom hole pressure) of 500 psi. The densities of water and oil at standard conditions are 64 lb/cu.ft. and 49 lb/cu.ft., respectively, and the viscosities are $\mu_o = 1.0$ cp, $\mu_w = 0.3$ cp. The fractional flow curve for this problem is shown in Figure 4.4. We see that flow is cocurrent for $0 \le S_w \le 0.38$ and countercurrent for $0.38 \le S_w \le 1$. The forward CFL number, $\max_{S\in[0,1]} f'_w$, is 3.73, whereas the backward CFL number, $-\min_{S\in[0,1]} f'_w$, is 0.638. We test our algorithm for uniform initial water saturations of $S_{wi} = 0.0, 0.1, \ldots, 0.9$. In each case, the simulation steps through $T = 1, 3, 7, 15, 30, 45, 60$ days (1 day $=$ 0.002 pore volumes), and afterwards the time-step size is fixed at $\Delta T = 20$ days until $T = 300$ days is reached, for a total of 21 steps. Table 4.1 shows the results for the standard and reduced Newton algorithms. We see that reduced Newton does not need to cut any time steps to achieve convergence, whereas standard Newton must cut the time step multiple times in four cases ($S_w = 0.0, 0.6, 0.7, 0.8$). Time-step cuts are very expensive, since it means that we must throw away the results of all previous iterations and start over. Moreover, the size of the next step following a time-step cut is usually set to the last successfully integrated Δt, i.e., the one reduced by the time-step cut. This can lead to a significantly smaller average time step size for a given simulation. Thus, a more stable algorithm that avoids time-step cuts can significantly outperform one that cuts time steps frequently, especially if their convergence rates are otherwise comparable. Table 4.1 shows that when neither algorithm requires time-step cuts, standard Newton converges more quickly some of the time ($S_w = 0.4, 0.5, 0.9$), whereas reduced Newton is quicker at other times ($S_w = 0.1, 0.2, 0.3, 0.6$). Nevertheless, the difference in average iteration count is less than 0.67 iterations per time step in all cases, so the convergence rates for both algorithms are comparable when no time-step cuts are needed. As we observe in later examples, the enhanced stability of reduced Newton does translate into gains in the overall run time for larger problems. The primary goal of this example is to demonstrate the robustness of reduced Newton, even in the presence of strong countercurrent flow. This property is essential if the algorithm is to be used in heterogeneous reservoirs with complicated permeability/porosity fields, especially since countercurrent flow due to gravity can be important in regions where the total velocity is small.

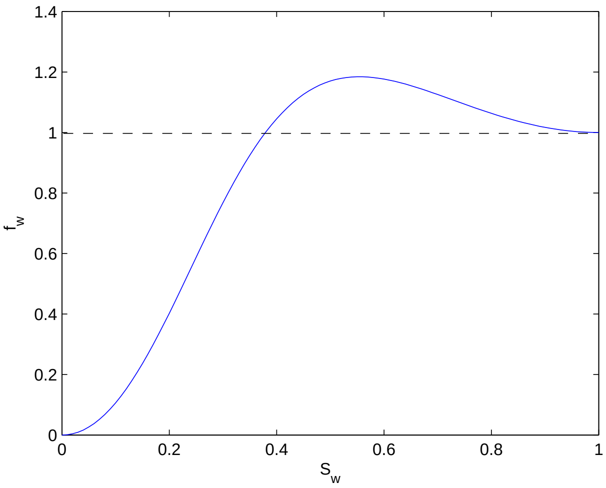


Figure 4.4: Fractional flow f_w for the 1D gravity example.

Table 4.1: Convergence history for 1D water floods with different initial water saturations. For both methods, Time steps = total number of time steps taken to simulate up to 300 days; Newtons = number of Newton iterations (excluding iterations wasted due to time-step cuts); Cuts = number of times the algorithm must cut the time-step size by half due to non-convergence.

	Standard			Reduced		
S_{wi}	Time steps	Newtons	Cuts	Time steps	Newtons	Cuts
0.0	26	140	5	21	61	0
0.1	21	59	0	21	58	0
0.2	21	59	0	21	58	0
0.3	21	50	0	21	49	0
0.4	21	51	0	21	58	0
0.5	21	67	0	21	81	0
0.6	22	88	2	21	85	0
0.7	24	96	6	21	90	0
0.8	23	85	3	21	84	0
0.9	21	51	0	21	65	0

4.3.2 Heterogeneous example with gravity

To demonstrate the effectiveness of reduced Newton on a large, complex heterogeneous reservoir, we test it on a water flood problem using a $2 \times 2 \times 2$ upscaling of the SPE 10 model [19]. This gives rise to a model with 141900 grid blocks ($110 \times 30 \times 43$). The reservoir model is shown in Figure 4.5. The top 18 layers of the reservoir represent a Tarbert formulation with highly variable permeabilities ranging from 4.8×10^{-3} to 1.2×10^3 md. The bottom 25 layers consist of an Upper Ness sequence, which is highly channelized. The porosity is 0.3 throughout the reservoir. Water is injected at the center of the reservoir at 5000 bbl/day (= 0.0002 pore volumes per day); four production wells are located in the four corners of the reservoir, operating at a bottom hole pressure of 4000 psi. Quadratic relative permeabilities are used with a residual saturation of 0.2 for both phases, and the viscosity ratio is 10. The rest of the parameters are the same as those in the original specification [19]. The simulation is carried out up to $T = 500$ days, which corresponds to 0.1 pore volumes injected (PVI). For any time step, if the global nonlinear solver does not converge within 20 iterations, the iterations are stopped, and the current time step is cut in half before restarting. Table 4.2 shows the convergence history of the standard and reduced Newton methods for an initial time step of 0.1 days. Here the time stepping is gentle enough that standard Newton does not need to cut time steps in order to achieve convergence. We see that reduced Newton takes fewer iterations than standard Newton to converge, and that the running time decreases from 728.6 seconds to 560.6 seconds. Thus, the savings from reducing the number of Newton iterations are more than enough to offset the cost of univariate solves.

Next we specify an initial time step of 1 day and track the number of Newton iterations required to converge. Figure 4.6 shows the results. We see that reduced Newton converges for the first time step in 9 iterations, whereas standard Newton does not converge and needs to cut the time step twice to converge with an initial time step of 0.25 days. Beyond the first time step, reduced Newton always takes fewer iterations to converge than its standard counterpart, and the iteration count does not exhibit the large variations that standard Newton does at the beginning.

Table 4.2: Convergence history for the upscaled SPE 10 model with an initial time step of 0.1 days. N = Number of Nonlinear (Newton) iterations; L = Number of Linear (CPR) solves; CFL = Maximum CFL number in the reservoir; %CC = Percentage of cell interfaces that experience countercurrent flow.

	Standard		Reduced			
days	N	L	N	L	CFL	%CC
0.1	4	18	4	17	1.8	6.2
0.3	3	17	3	17	1.9	2.4
0.7	3	18	2	12	2.1	1.1
1.5	3	19	2	14	2.5	0.7
3.1	4	26	2	15	4.0	0.5
6.3	5	32	2	16	6.7	0.5
10	4	26	2	15	11.1	0.5
20	6	45	3	27	23.9	0.5
35	4	32	3	27	35.2	0.5
50	3	27	2	19	33.2	0.5
70	4	35	3	27	35.1	0.6
90	4	33	3	28	35.6	0.6
110	4	37	3	30	52.9	0.6
140	4	41	3	34	112.1	0.6
170	4	39	2	21	102.8	0.7
200	4	35	2	21	145.3	0.7
230	3	33	2	22	129.1	0.7
260	3	33	2	22	132.0	0.8
290	3	30	2	21	132.3	0.8
320	3	31	2	21	119.6	0.8
350	3	30	2	19	109.5	0.8
380	3	30	2	20	116.7	0.9
410	3	31	2	20	112.0	0.9
440	3	30	2	19	114.9	0.9
470	3	28	2	19	108.1	1.0
500	3	29	2	19	146.3	1.0
Total	93	785	61	542		
Running time (s)	728.6		560.6			

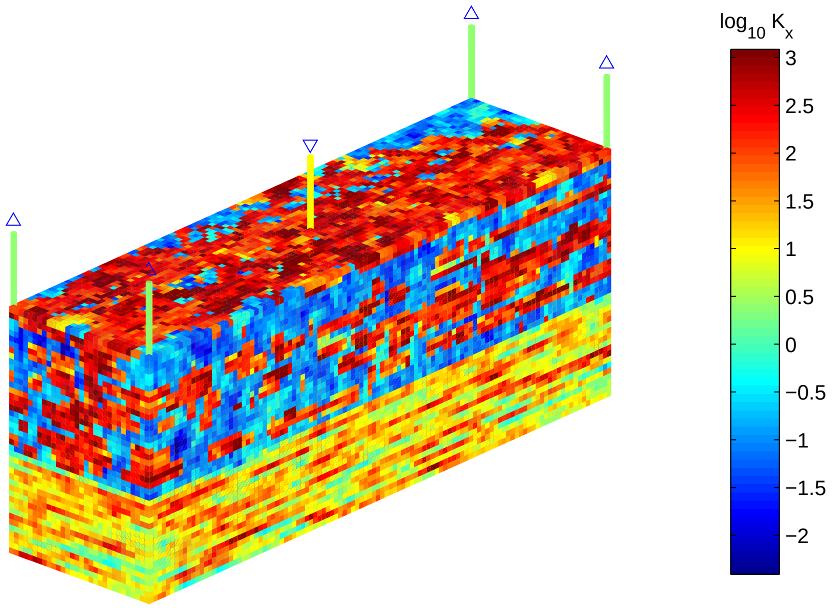


Figure 4.5: Permeability field and well configuration for the upscaled SPE 10 problem[19]. The reservoir is displayed upside down so that the channels in the bottom layers are clearly visible.

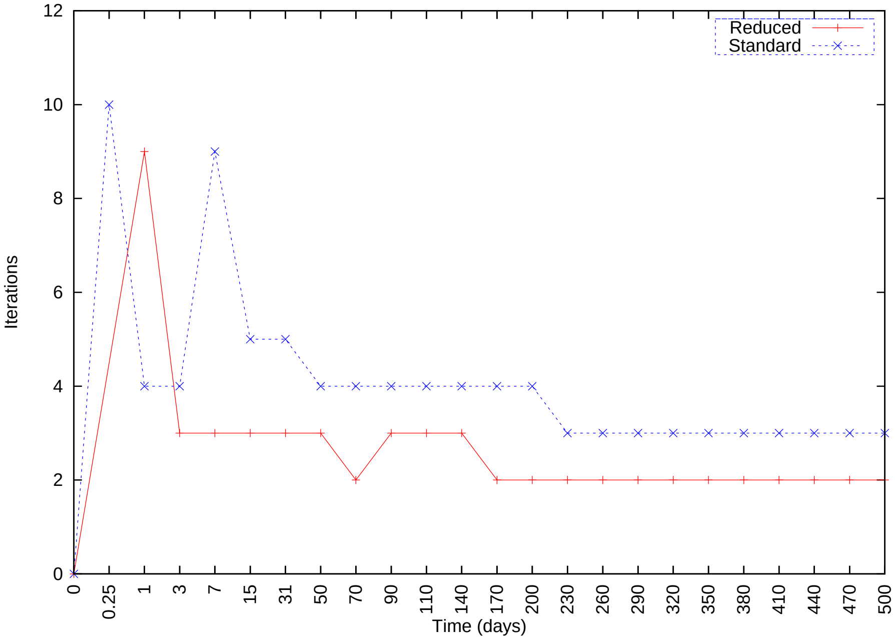


Figure 4.6: Convergence history for the upscaled SPE 10 model with an initial time step of 1 day.

4.3.3 Large heterogeneous example

Since the cost of the single cell solves scales linearly with the problem size, we expect that the savings from the potential-based reduced Newton method will become even more evident in large heterogeneous examples, where the computational cost is dominated by the solution of linear systems. We demonstrate this by simulating the full SPE 10 problem ($60 \times 220 \times 85 = 1.12$ million grid blocks) and with the variable porosity field as specified in [19]. The published relative permeabilities and fluid properties are used, except that the formation volume factor B_o and the density ρ_o are taken to be the same as the published B_w and ρ_w. The injection rate is 5000 bbl/day (0.000366 pore volumes per day). The simulation runs until $T = 2000$ days ($PVI = 0.732$). Three time-stepping strategies are used:

- Short time steps: $T = 0.01$, 0.03, 0.07, 0.15, 0.31, 0.63, 1, 3, 7, 15, 31, 63, 90, 120, 150, 180, 220, 260, 300 days. After 300 days, $\Delta T = 50$ days (0.0183 pore volumes) until $T = 2000$ days is reached.

- Long time steps: $T = 0.01$, 0.31, 1, 7, 31, 90, 150, 220, 300 days. After 300 days, $\Delta T = 100$ days (0.0366 pore volumes) until $T = 2000$ days is reached.

- Huge time steps: $T = 0.01$, 0.31, 1, 7, 31, 90, 200 days. After 200 days, $\Delta T = 500$ days (0.183 pore volumes) until $T = 2000$ days is reached.

As before, the time step is cut in half if the global nonlinear solver does not converge within 20 iterations. Table 4.3 summarizes the runs for both the standard and reduced Newton algorithms, and Figure 4.7 compares the convergence histories of standard and reduced Newton for the long time step case. We observe that reduced Newton can easily handle the “long” and “huge” time step cases. Standard Newton, on the other hand, needs to cut time steps multiple times in order to achieve convergence, and this results in a significant number of wasted linear solves and a serious degradation in performance. In fact, we were unable to run standard Newton for the huge time step case because of the large number of time step cuts. Consistent with the collective experience in the simulation community, taking too large a time step in standard Newton actually makes the simulation slower. The opposite is true for

Table 4.3: Summary of runs for the full SPE 10 problem. “Wasted Newton steps” and “wasted linear solves” indicate the number of Newton iterations and linear solves that are wasted because of time step cuts.

	Standard		Reduced		
	Short Δt	Long Δt	Short Δt	Long Δt	Huge Δt
No. of time steps	58	38	53	26	11
No. of time step cuts	6	17	0	0	0
No. of Newton steps	353	516	128	90	55
– Wasted Newton steps	120	340	0	0	0
No. of linear solves	3818	6257	2271	2399	1805
– Wasted linear solves	860	3934	0	0	0
Total running time (sec)	24053	37388	16558	14727	10275
– Linear solves (sec)	22570	35457	11697	11301	7899
– Single-cell solves (sec)	0	0	4194	2996	2132

reduced Newton. Indeed, reduced Newton with long or huge time steps runs in less than 60% of the time required by standard Newton with either time-stepping strategy. Finally, Figure 4.8 shows the oil production rate and water cut for all four simulation runs. The discrepancy between the solutions is insignificant, with the exception of the huge time step case, in which the time truncation error becomes so large that the water cut and production curves noticeably deviate from the cases. In practice, one would probably not want to take such a large time step, but it is reassuring to know that reduced Newton can still converge under such extreme circumstances. In general, by using reduced Newton with (reasonably) larger time steps, we obtain substantial speedups with little or no change in solution accuracy.

4.3.4 1D three-phase example with gravity

To show that the reduced formulation is applicable to three-phase flow, the algorithm is tested on a three-phase model in which gas is injected into a reservoir initially saturated with a mixture of 50% oil and 50% water in every cell. This saturation is chosen to ensure that all phases are mobile, and that we have a truly three-phase

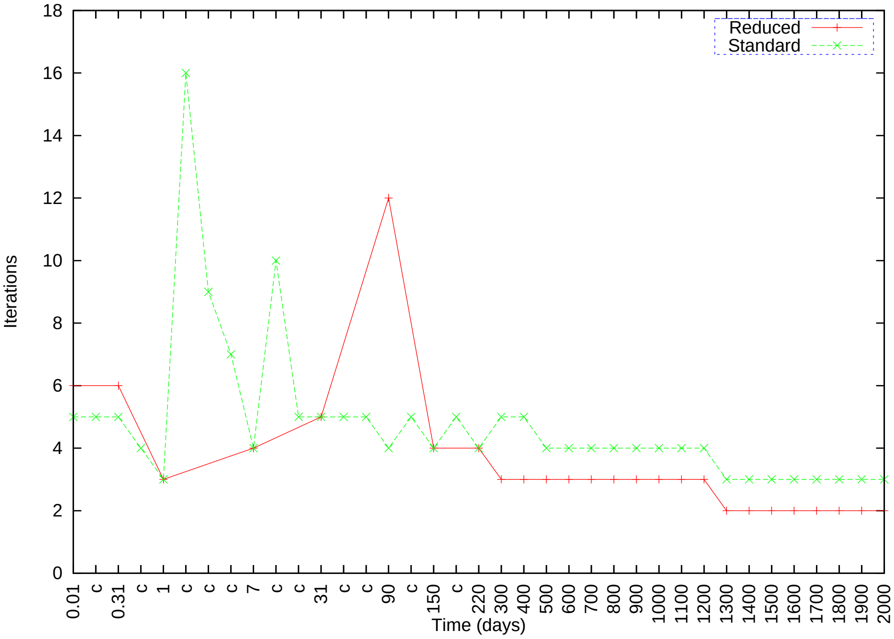


Figure 4.7: Convergence history for the full SPE 10 problem with long time steps. Tick marks on the x-axis labeled c correspond to intermediate time steps needed by standard Newton to achieve convergence; these steps are skipped by reduced Newton.

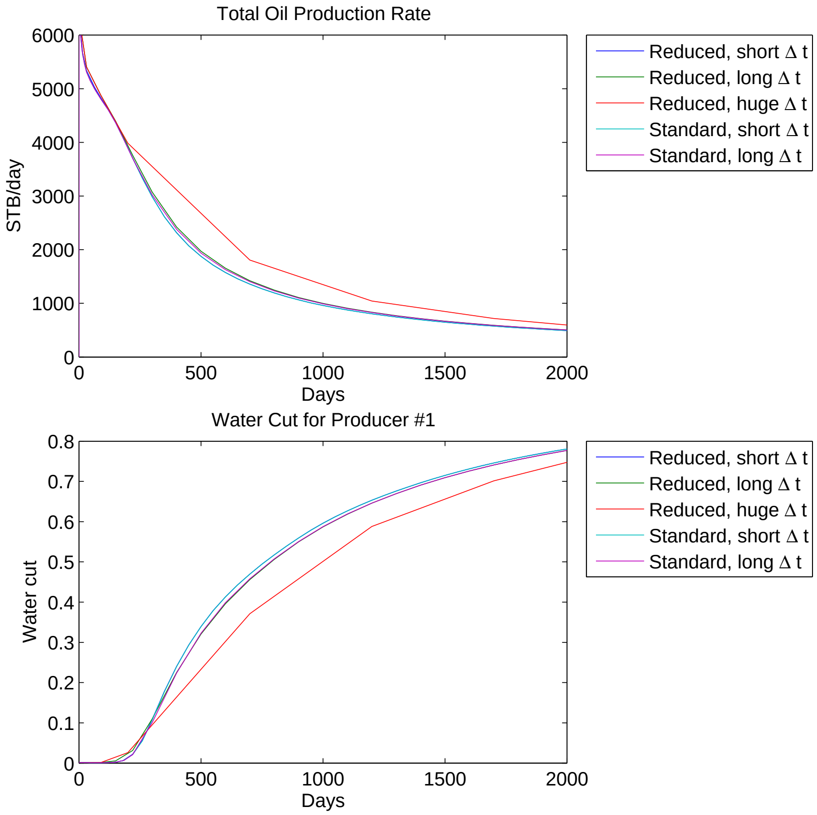


Figure 4.8: Total oil production rate and water cut for the full SPE 10 problem.

problem. The reservoir is identical to the one used in Example 4.3.1. The PVT data and relative permeabilities are shown in Tables 4.4 and 4.5, respectively. For simplicity, the gas component is assumed not to dissolve into the oil phase (i.e., $R_{go} = 0$). The oil relative permeability is interpolated from the oil-gas and oil-water tables using the Stone I method. Gas is injected into the top layer at a rate of 100 MSCF/day (0.000768 pore volumes/day at 4000 psi), and a producer in the bottom layer is maintained at a constant pressure of 4000 psi. The production curve is shown in Figure 4.9. Even though gas is highly mobile ($\mu_w/\mu_g = 11.6$, $\mu_o/\mu_g = 111.9$), breakthrough occurs relatively late (at $T = 521$ days or 0.4 pore volumes) because gas preferentially stays in the upper layers because of buoyancy. In addition, since the simulation does not start from gravity equilibrium, gravity segregation between oil and water must occur at the initial stages of the simulation. Up to 98% of cell interfaces experience countercurrent flow at some point before gas breaks through. This accounts for the rather complicated behavior of the water and oil production curves prior to gas breakthrough. Even though this is a rather small example, we believe it captures the essence of the types of nonlinearity present in countercurrent three-phase flow, and provides a good test case for comparing the convergence behavior of the standard and reduced Newton algorithms. In this example, two time-stepping strategies are used:

- Short time steps: $T = 0.1, 1, 5, 10$ days. After 10 days, $\Delta t = 10$ days (0.00768 pore volumes) until $T = 1000$ days.

- Long time steps: After an initial time step of 0.1 days, Δt is automatically chosen based on saturation and pressure changes, with a minimum of $\Delta t = 10$ days and gradually increasing until $\Delta t = 100$ days (0.0768 pore volumes).

Table 4.6 summarizes the runs for the standard and potential-based reduced Newton algorithms. Running times have little meaning because of the small size of the problem, and are thus omitted. We once again observe that reduced Newton has no difficulty handling both short and long time steps, whereas standard Newton needs to cut the time-step size repeatedly throughout the simulation. Thus, the presence of three phases does not negatively impact the convergence behavior of reduced Newton.

Table 4.4: PVT relations for all three-phase examples.

P (psi)	B_o (RB/STB)	μ_o (cp)	B_w (RB/STB)	μ_w (cp)	B_g (RB/SCF)	μ_g (cp)
14.7	1.062	2.200	1.0410	0.31	0.166666	0.0080
264.7	1.061	2.850	1.0430	0.31	0.012093	0.0096
514.7	1.060	2.970	1.0395	0.31	0.006274	0.0112
1014.7	1.059	2.990	1.0380	0.31	0.003197	0.0140
2014.7	1.056	2.992	1.0350	0.31	0.001614	0.0189
2514.7	1.054	2.994	1.0335	0.31	0.001294	0.0208
3014.7	1.053	2.996	1.0320	0.31	0.001080	0.0228
4014.7	1.050	2.998	1.0290	0.31	0.000811	0.0268
5014.7	1.047	3.000	1.0258	0.31	0.000649	0.0309
9014.7	1.033	3.008	1.0130	0.31	0.000386	0.0470

Table 4.5: Relative permeabilities for all three-phase examples.

S_w	k_{rw}	k_{row}	S_g	k_{rg}	k_{rog}
0.12	0	1.00	0	0	1.00
0.121	1.67E-12	1.00	0.001	0.0002	1.00
0.14	2.67E-07	0.997	0.02	0.0033	0.997
0.17	1.04E-05	0.98	0.05	0.0106	0.98
0.24	3.46E-04	0.7	0.12	0.0364	0.70
0.32	2.67E-03	0.35	0.20	0.0919	0.35
0.37	6.51E-03	0.2	0.25	0.1459	0.20
0.42	0.014	0.09	0.30	0.2226	0.09
0.52	0.043	0.021	0.40	0.4588	0.021
0.57	0.068	0.01	0.45	0.6336	0.01
0.62	0.104	0.001	0.50	0.7449	0.001
0.72	0.216	0.0001	0.60	0.8887	0.0001
0.82	0.400	0	0.70	0.9563	0
1.00	1.000	0	0.88	1.0000	0

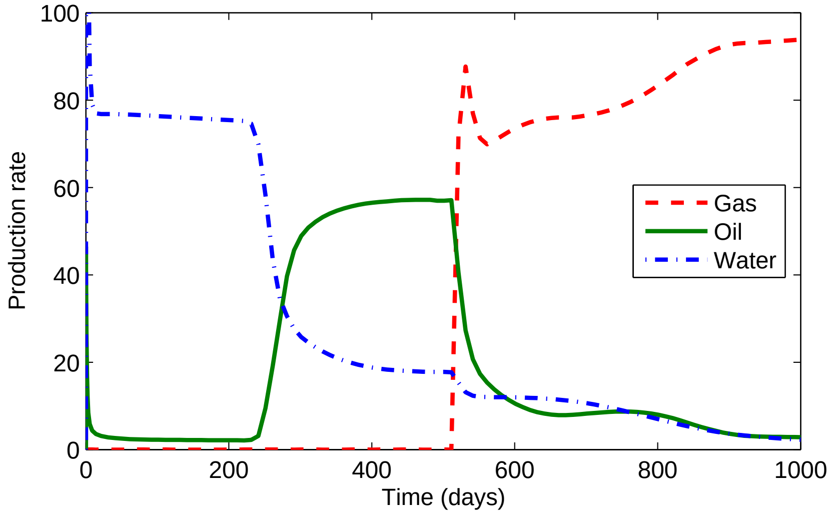


Figure 4.9: Production curve for the 1D three-phase example. The units are STB/day for oil and water, and MSCF/day for gas.

Table 4.6: Summary of runs for the 1D three-phase example with gravity. “Wasted Newton steps” and “wasted linear solves” indicate the number of Newton iterations and linear solves that are wasted because of time step cuts.

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
No. of time steps	111	74	103	26
No. of time step cuts	16	36	0	0
No. of Newton steps	888	1223	480	229
– Wasted Newton steps	320	720	0	0
No. of linear solves	1763	2421	973	480
– Wasted linear solves	641	1418	0	0

4.3.5 2D Heterogeneous three-phase example

We now test the reduced Newton algorithm on a three-phase example with heterogeneity. The reservoir consists of the 51st layer of the SPE 10 problem, which is a slice in the Upper Ness formation (see Example 4.3.3). Initially the reservoir contains a mixture of 50% oil and 50% water, and gas is injected through a well in the center at a rate of 1000 MSCF/day (0.00005 pore volumes per day). The four production wells (one in each corner) are each maintained at a bottom hole pressure of 4000 psi. The PVT and relative permeability data are the same as in Example 4.3.4 and are given in Tables 4.4 and 4.5. The simulation is run up to $T = 500$ days (0.025 PVI), which is much larger than the breakthrough time ($T_{BT} \approx 40$ days or 0.002 PVI). Note that the early breakthrough time is due to the extremely high mobility of the gas. Figure 4.10 shows the gas saturation of the reservoir at $T = 500$ days. Two time-stepping strategies are used:

- Short time steps: $T = 1, 3, 7, 15, 31, 63, 100$ days. After 100 days, $\Delta t = 50$ days (0.00125 pore volumes) until $T = 500$ days is reached.
- Long time steps: $T = 10, 30, 60, 100$ days. After 100 days, $\Delta t = 100$ days (0.0025 pore volumes) until $T = 500$ days is reached.

Table 4.7 shows the performance of the standard and reduced Newton algorithms. Once again no time step cuts are required by reduced Newton, demonstrating its stability compared with the standard Newton's method. This translates to an improvement in running time for the long time step case. This example shows that the improvement obtained from reduced Newton in three-phase flow is not limited to simple 1D cases.

4.3.6 3D three-phase example

Here, the algorithm is tested on a 3D three-phase model in which gas is injected into a reservoir containing a mixture of 50% oil and 50% water. The reservoir ($20 \times 20 \times 3$ cells) is a 2×2 areal refinement of the one used in the SPE1 test set [57] and is shown in Figure 4.11. The PVT data and relative permeabilities are the same as the two

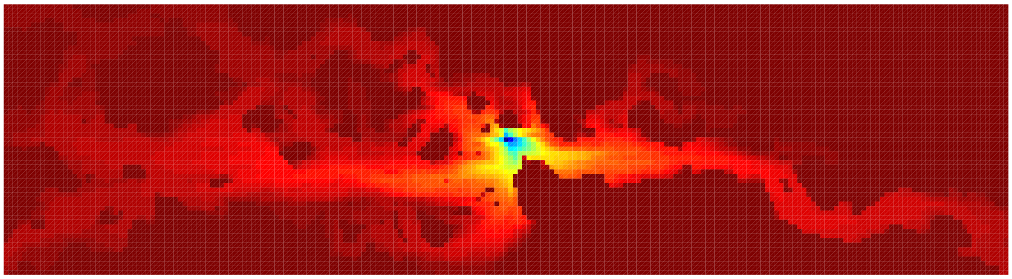


Figure 4.10: Gas saturation at $T = 500$ days in the 2D heterogeneous three-phase example. Dark blue indicates 100% gas, whereas dark red indicates a cell consisting purely of liquid phases.

Table 4.7: Summary of runs for the 2D heterogeneous three-phase example. "Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted because of time step cuts.

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
No. of time steps	16	10	15	8
No. of time step cuts	1	3	0	0
No. of Newton steps	74	101	58	40
– Wasted Newton steps	20	60	0	0
No. of linear solves	1264	1529	1172	881
– Wasted linear solves	276	698	0	0
Total running time (sec)	63.5	75.6	73.8	53.9
– Linear solves (sec)	53.7	66.1	50.5	37.9
– Single-cell solves (sec)	0	0	18.6	13.0

previous examples (Table 4.4 and 4.5), and the Stone I model is used to interpolate the oil-gas and oil-water data. The gas-injection well is completed in cell (1,1,1) and operates at 100000 MSCF/day (0.000073 pore volumes per day at 9000 psi); a production well, completed in cell (20,20,3), operates at a bottom-hole pressure of 1000 psi. The simulation is run up to T = 5000 days (0.365 PVI). Because of the high gas mobility, breakthrough occurs very early ($T_{BT} \approx$ 100 days or 0.0073 PVI). Since the oil and water are not in gravity equilibrium at the start of the simulation, there is significant countercurrent flow in the problem. Two time-stepping strategies are used:

- Short time steps: $T = 30, 100, 200, 250, 400, 600, 900$ days. After 900 days, Δt = 400 days (0.0292 pore volumes) until T = 5000 days.
- Long time steps: $T = 100, 250, 600$ days. After 600 days, Δt = 800 days (0.0584 pore volumes) until T = 5000 days.

Table 4.8 shows the performance of the standard and reduced Newton algorithms. Again we see that the reduced Newton method requires no time step cuts and fewer iterations to converge compared to the standard Newton method.

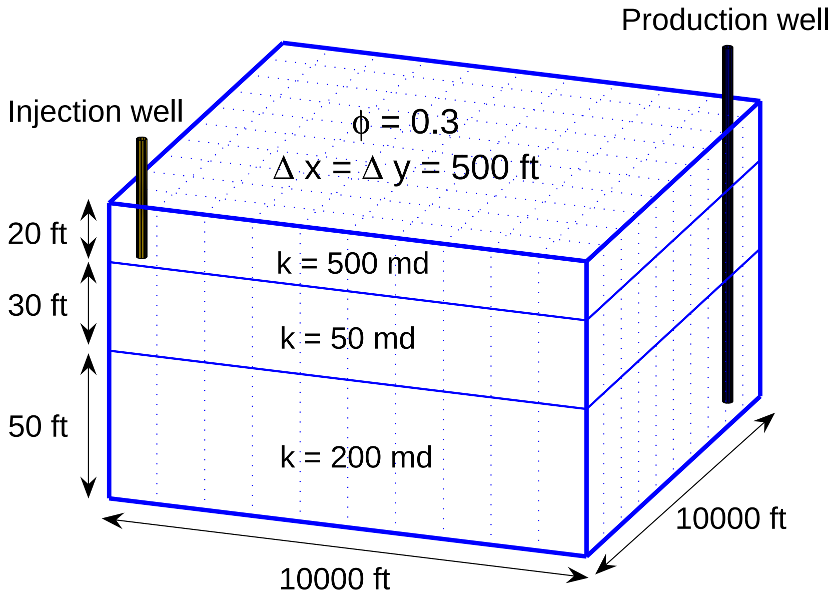


Figure 4.11: Reservoir description for the 3D three-phase example.

Table 4.8: Summary of runs for 3D three-phase example. "Wasted Newton steps" and "wasted linear solves" indicate the number of Newton iterations and linear solves that are wasted because of time step cuts.

	Standard		Reduced	
	Short Δt	Long Δt	Short Δt	Long Δt
No. of time steps	18	11	17	9
No. of time step cuts	1	3	0	0
No. of Newton steps	95	117	74	57
– Wasted Newton steps	20	60	0	0
No. of linear solves	1083	1624	974	838
– Wasted linear solves	178	855	0	0
Total running time (sec)	4.9	6.5	6.1	4.9
– Linear solves (sec)	3.7	5.5	3.4	2.9
– Single-cell solves (sec)	0	0	2.1	1.6

Chapter 5

Linear Preconditioning

When an immiscible n_p-phase flow problem is discretized on a grid containing N cells, Newton's method requires the solution of a sparse $n_pN \times n_pN$ linear system $Jx = r$ at every iteration. The matrix J, which comes from the linearized residual functions

$$\frac{V_i\phi_i}{\Delta t}(S_{p,i}^{n+1} - S_{p,i}^{n}) + \sum_{l\in\mathrm{adj}(i)} |\partial V_{il}| F_{p,il}(S,p) = q_{p,i}, \tag{5.0.1}$$

inherits the mixed hyperbolic-parabolic character of the underlying PDEs, which means methods developed for a specific type of discretized PDE (e.g., elliptic PDEs) will not work well for J. For this reason, efficient solution of the linear systems remains a challenging problem in reservoir simulation. Direct solvers become prohibitively expensive as the grid is refined; this is especially true for 3D problems, where LU factorization requires $O(N^2)$ floating-point operations and $O(N^{4/3})$ storage, even when an optimal ordering strategy such as nested dissection is used [38]. On the other hand, when iterative methods are used, standard preconditioners such as incomplete LU factorizations and multigrid perform poorly because the problem is neither purely hyperbolic nor purely elliptic.

It is well known that the ordering of equations and unknowns can have a huge impact on the quality of various preconditioners [25, 30, 10]. In most of these works the orderings considered tend to belong to the following categories:

1. Coloring-based orderings, in which the nodes in the adjacency graph are partitioned into a finite number of colors, and nodes with the same color are ordered within the same block. The red-black ordering is a classical example of such orderings, which are often motivated by parallelization considerations or in the context of cyclic reduction.

2. Fill-minimizing orderings, which are developed in the context of sparse direct solvers in order to minimize the number of fill-in entries in the LU factorization. Examples include the reverse Cuthill-McKee method and minimum degree ordering [38].

The above ordering strategies, while having the advantage of being applicable to general sparse matrices, do not exploit the underlying physics of the problem. For advection dominated problems, a natural idea is to order the cells according to flow direction (e.g., from upstream to downstream). Ordering of this type has been considered in the CFD community (cf. [54]), but its use is limited in reservoir simulation. The aim of this chapter is to exploit the cell-based and phase-based orderings introduced in Chapter 3 for preconditioning purposes. In particular, we proceed as follows:

1. Propose an improvement to the standard CPR-BILU(0) preconditioner that exploits cell-based ordering;

2. Use phase-based ordering to derive preconditioned Krylov solvers based on Schur complement preconditioning.

5.1 Structure of the Jacobian matrix

In this chapter, we mainly consider Jacobians that arise from a fully implicit, five-point finite-volume discretization of the incompressible black-oil equations, with upstream weighting for saturation-dependent terms. When phase-based ordering is used, the Jacobian will be denoted by J, which has the form (cf. (3.2.3),(3.2.5))

$$J = \begin{array}{cc} S_w & p \\ \left[\begin{array}{c} J_{ss} \\ J_{ps} \end{array}\right. & \left.\begin{array}{c} J_{sp} \\ J_{pp} \end{array}\right] \end{array} \quad \begin{array}{l} \text{water equation} \\ \text{oil equation} \end{array} \tag{5.1.1}$$

In addition, we will often use cell-based ordering, in which all the equations and variables belonging to the same control volume are grouped into a single block. In this case, the Jacobian is denoted by A, where

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{bmatrix} \tag{5.1.2}$$

is a block matrix with $n_p \times n_p$ blocks. Each block row represents the derivatives of the conservation equations (oil and water) with respect to the discrete unknowns (S_i and p_i) at the gridblock and its adjacent cells. For example,

$$A_{ii} = \begin{bmatrix} (J_{ss})_{ii} & (J_{sp})_{ii} \\ (J_{ps})_{ii} & (J_{pp})_{ii} \end{bmatrix}.$$

Clearly, $A = PJP^T$ for some permutation matrix P. For simplicity, we assume two-phase flow throughout this chapter, while noting that many results can be extended to three-phase flow. We make the following additional assumptions.

Assumptions 5.

1. The phase mobilities are non-negative and satisfy $\lambda_w' = \partial\lambda_w/\partial S_w > 0$ and $\lambda_o' = \partial\lambda_o/\partial S_w < 0$;

2. The total mobility $\lambda_t = \lambda_o + \lambda_w$ across each cell boundary is strictly positive;

3. Phase-based upstreaming is performed based on the upstream directions given at the linearization point (P^ℓ, S^ℓ);

4. A pressure Dirichlet boundary condition is prescribed on a segment of the boundary with positive measure. (Alternatively, one can assume there exists at least one production well operating at a fixed bottom-hole pressure.)

Assumption 5.1 has already been stated in Theorem 2.1. Assumption 5.2 is similar to the uniform ellipticity condition in Section 4.2. When the flow is cocurrent, it is purely an assumption on the fluid mobilities. In the countercurrent flow case, however, it is also an assumption on the linearization point (S^ℓ, P^ℓ), since it is possible that λ_w and λ_o are evaluated at two different saturations because of upstreaming. Thus, if there are adjacent cells i and $i+1$ such that $S_i^\ell = 0$ and $S_{i+1}^\ell = 1$, then Assumption 5.2 would disallow the possibility that the upstream directions for water and oil are i and $i+1$ respectively, which is essentially a restriction on the set of admissible pressure profiles P^ℓ. Assumption 5.3 ensures monotonicity of the discretization (in the sense of Chapter 2), and assumption 5.4 is needed for a unique pressure solution.

Lemma 5.1. *Assume the hypothesis given in Assumptions (5.1–4). Then the sub-blocks of the Jacobian J have the following properties:*

1. *$J_{ss} = (1/\Delta t)D + J_{ss}^0$ and $J_{ps} = -(1/\Delta t)D + J_{ps}^0$, where D is a positive diagonal matrix, and J_{ss}^0 and $-J_{ps}^0$ are weakly column diagonally-dominant M-matrices;*

2. *J_{sp} and J_{pp} are weakly diagonally dominant, symmetric, positive semi-definite matrices;*

3. *$J_{sp} + J_{pp}$ is a symmetric, positive-definite, irreducibly diagonally dominant M-matrix;*

Moreover, the matrices J_{ss}^0, J_{ps}^0, J_{sp} and J_{pp} are independent of Δt.

Based on the above lemma, the following theorems concerning the rank of J can be proven. Clearly, we have

$$J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{ps} & J_{pp} \end{bmatrix} \text{ nonsingular } \iff \tilde{J} = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{ts} & J_{tp} \end{bmatrix} \text{ nonsingular,}$$

where $J_{ts} = J_{ss} + J_{ps}$ and $J_{tp} = J_{sp} + J_{pp}$. That is, J_{ts} and J_{tp} are the Jacobian matrices corresponding to the total mass balance equation (1.1.19).

Since J_{tp} is nonsingular, $\tilde{J}$ is nonsingular if and only if the *Schur complement*

$$S_1 := J_{ss} - J_{sp}J_{tp}^{-1}J_{ts}$$

is also nonsingular.

Theorem 5.2. *There exists $T > 0$ such that J is nonsingular for $0 < \Delta t < T$.*

Proof. First, note that $J_{ts} = J_{ss} + J_{ps} = J_{ss}^0 + J_{ps}^0$ is independent of Δt. Thus, we can write

$$S_1 = \frac{1}{\Delta t}D + (J_{ss}^0 - J_{sp}J_{tp}^{-1}J_{ts}),$$

where the terms in brackets are independent of Δt. Now S_1 is nonsingular if and only if

$$\Delta t D^{-1}S_1 = I + \Delta t D^{-1}(J_{ss}^0 - J_{sp}J_{tp}^{-1}J_{ts})$$

is also nonsingular, which is the case whenever

$$\rho(\Delta t D^{-1}(J_{ss}^0 - J_{sp}J_{tp}^{-1}J_{ts})) < 1,$$

where $\rho(\cdot)$ denotes the spectral radius. Thus, S_1 is nonsingular whenever $0 < \Delta t < T$, where

$$T = \frac{1}{\rho(D^{-1}(J_{ss}^0 - J_{sp}J_{tp}^{-1}J_{ts}))},$$

or $T = \infty$ if $\rho(D^{-1}(J_{ss}^0 - J_{sp}J_{tp}^{-1}J_{ts})) = 0$. □

Theorem 5.3. *For 1D flow problems, J is nonsingular for all $\Delta t > 0$.*

Proof. For 1D problems, it is possible to write explicitly down the form of the Schur complement S_1 by eliminating the pressure terms directly. Since the discretization and linearization steps commute, it is notationally more convenient to manipulate the PDE itself, although one can also perform the same calculation on the discrete equations. We start with the two-phase conservation law:

$$\phi S_t - \frac{\partial}{\partial x}\big[\lambda_w(p_x + \rho_w g z_x)\big] = 0, \tag{5.1.3}$$

$$-\phi S_t - \frac{\partial}{\partial x}\big[\lambda_o(p_x + \rho_o g z_x)\big] = 0. \tag{5.1.4}$$

Linearize around (S^ℓ, P^ℓ) by letting $S = S^\ell + \sigma$, $p = P^\ell + \pi$:

$$\phi S_t^\ell - \frac{\partial}{\partial x}\big[\lambda_w(P_x^\ell + \rho_w g z_x)\big] + \left\{\phi\sigma_t - \frac{\partial}{\partial x}\big[\lambda_w'\sigma^w(P_x^\ell + \rho_w g z_x) + \lambda_w \pi_x\big]\right\} = 0, \tag{5.1.5}$$

$$-\phi S_t^\ell - \frac{\partial}{\partial x}\big[\lambda_o(P_x^\ell + \rho_o g z_x)\big] + \left\{-\phi\sigma_t - \frac{\partial}{\partial x}\big[\lambda_o'\sigma^o(P_x^\ell + \rho_o g z_x) + \lambda_o \pi_x\big]\right\} = 0. \tag{5.1.6}$$

In the above equations, all coefficients are evaluated at the linearization point, so they do not depend on σ and π. We also used the notation σ^w and σ^o to denote the upwind direction of the water and oil phase in the finite volume discretization, which can be different in general. By keeping the terms separate we can easily mimic this manipulation in the discrete case. If we define

$$F(x,t) := -\phi S_t^\ell + \frac{\partial}{\partial x}\big[\lambda_w(P_x^\ell + \rho_w g z_x)\big], \tag{5.1.7}$$

$$G(x,t) := \phi S_t^\ell + \frac{\partial}{\partial x}\big[\lambda_o(P_x^\ell + \rho_o g z_x)\big], \tag{5.1.8}$$

we obtain the linearized PDE

$$\phi\sigma_t - \frac{\partial}{\partial x}\big[\lambda_w'\sigma^w(P_x^\ell + \rho_w g z_x) + \lambda_w \pi_x\big] = F(x,t), \tag{5.1.9}$$

$$-\phi\sigma_t - \frac{\partial}{\partial x}\big[\lambda_o'\sigma^o(P_x^\ell + \rho_o g z_x) + \lambda_o \pi_x\big] = G(x,t). \tag{5.1.10}$$

We can now eliminate π_x to obtain a single equation involving σ. Adding (5.1.9) and

(5.1.10) and integrating gives

$$\lambda_w'\sigma^w(P_x^\ell+\rho_w g z_x)+\lambda_o'\sigma^o(P_x^\ell+\rho_o g z_x)+\lambda_T\pi_x=-\int_0^x(F(\xi,t)+G(\xi,t))d\xi=:-H(x,t) \tag{5.1.11}$$

so that

$$\pi_x=-\frac{1}{\lambda_T}\left\{H(x,t)+\lambda_w'\sigma^w(P_x^\ell+\rho_w g z_x)+\lambda_o'\sigma^o(P_x^\ell+\rho_o g z_x)\right\}. \tag{5.1.12}$$

Substituting into (5.1.9) gives

$$\begin{aligned}\phi\sigma_t-\frac{\partial}{\partial x}\Bigg[\lambda_w'\sigma^w(P_x^\ell+\rho_w g z_x)-\frac{\lambda_w}{\lambda_T}\big(H(x,t)\\ +\lambda_w'\sigma^w(P_x^\ell+\rho_w g z_x)+\lambda_o'\sigma^o(P_x^\ell+\rho_o g z_x)\big)\Bigg]=F(x,t).\end{aligned} \tag{5.1.13}$$

Simplify and get

$$\phi\sigma_t-\frac{\partial}{\partial x}\left[\frac{\lambda_o\lambda_w'(P_x^\ell+\rho_w g z_x)}{\lambda_T}\sigma^w-\frac{\lambda_w\lambda_o'(P_x^\ell+\rho_o g z_x)}{\lambda_T}\sigma^o\right]=R(x,t), \tag{5.1.14}$$

where $R(x,t)$ is some combination of $F(x,t)$ and $H(x,t)$ that does not depend on σ and π, and hence is unimportant for the analysis. To derive the discrete form of (5.1.14), we need to resolve the upstreamed saturations σ^w and σ^o, which are given by

$$\sigma_{i+1/2}^w=\begin{cases}\sigma_{i+1}, & P_x^\ell+\rho_w g z_x\geq 0\\ \sigma_i, & P_x^\ell+\rho_w g z_x<0,\end{cases}$$

and similarly for σ^o. Thus, the discrete algebraic equations that arise from Newton's method are of the form

$$\frac{\phi_i\sigma_i}{\Delta t}+\frac{1}{\Delta x_i}\big[\alpha_{i+1/2}\sigma_i-\beta_{i+1/2}\sigma_{i+1}-\alpha_{i-1/2}\sigma_{i-1}+\beta_{i-1/2}\sigma_i\big]=R_i, \tag{5.1.15}$$

where

$$\begin{aligned}
\alpha_{i+1/2} &= \frac{\lambda_{w,i+1/2}\lambda'_{o,i}\min\{P^\ell_{i+1} - P^\ell_i + \rho_o g\Delta z, 0\}}{\Delta x_{i+1/2}(\lambda_{o,i+1/2} + \lambda_{w,i+1/2})} \\
&\quad - \frac{\lambda_{o,i+1/2}\lambda'_{w,i}\min\{P^\ell_{i+1} - P^\ell_i + \rho_w g\Delta z, 0\}}{\Delta x_{i+1/2}(\lambda_{o,i+1/2} + \lambda_{w,i+1/2})} \geq 0, \\
\beta_{i+1/2} &= -\frac{\lambda_{w,i+1/2}\lambda'_{o,i+1}\max\{P^\ell_{i+1} - P^\ell_i + \rho_o g\Delta z, 0\}}{\Delta x_{i+1/2}(\lambda_{o,i+1/2} + \lambda_{w,i+1/2})} \\
&\quad + \frac{\lambda_{o,i+1/2}\lambda'_{w,i+1}\max\{P^\ell_{i+1} - P^\ell_i + \rho_w g\Delta z, 0\}}{\Delta x_{i+1/2}(\lambda_{o,i+1/2} + \lambda_{w,i+1/2})} \geq 0.
\end{aligned}$$

Thus, with proper scaling, the Schur complement S_1 has the form

$$S_1 = \begin{bmatrix}
\gamma_1 + \alpha_{3/2} + \beta_{1/2} & -\beta_{3/2} & & & \\
-\alpha_{3/2} & \gamma_2 + \alpha_{5/2} + \beta_{3/2} & \ddots & & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & -\beta_{N-1/2} \\
& & & -\alpha_{N-1/2} & \gamma_N + \alpha_{N+1/2} + \beta_{N-1/2}
\end{bmatrix},$$

where $\gamma_i = \Delta x_i \phi_i / \Delta t$. This is a tridiagonal, column diagonally dominant M-matrix, which means S_1 is nonsingular (and in fact positive-stable) whenever $\Delta t > 0$. Hence, the full Jacobian J is also nonsingular. □

We remark that, in most practical reservoir flow simulations, it is extremely rare to encounter a singular Jacobian unless the linearization point (S^ℓ, P^ℓ) is so far from the solution that it is physically inadmissible (e.g., when P^ℓ no longer satisfies the maximum principle).

5.2 CPR preconditioning

One of the most successful approaches for preconditioning fully-implicit Jacobians is the two-stage constrained pressure residual (CPR) method proposed by Wallis [81]. The method can be viewed as the linear analog of the sequential implicit (SEQ) method, in the sense that it first decouples the full problem into an elliptic and a hyperbolic subproblem; then at each iteration, one would first solve the elliptic problem to obtain an approximate pressure, and then use this pressure to solve the transport problem. A more precise description in terms of two-stage preconditioners follows.

For a linear system $Jx = r$, the general two-stage preconditioner is given by

$$M^{-1} = T_2\big[I - JT_1\big] + T_1, \tag{5.2.1}$$

where T_1 and T_2 are approximate inverses for J, or for the restriction of J onto some subspace. When T_1 and T_2 are both invertible, then M^{-1} is equivalent to the preconditioner derived from the two-stage stationary iteration

$$\begin{aligned} T_1^{-1}x^{k+1/2} &= (T_1^{-1} - J)x^k + b, \\ T_2^{-1}x^k &= (T_2^{-1} - J)x^{k+1/2} + b. \end{aligned}$$

Examples of this type include ADI preconditioners [68], the symmetric SOR method [39] and the HSS method [7]. The T_i can be singular as well. The special case of

$$T_i = R_i(R_i^T A R_i)^{-1} R_i^T,$$

where R_i^T is a restriction operator, corresponds to either a block Gauss-Seidel or a multiplicative Schwarz method, depending on whether the blocks overlap (cf. [68]). Since

$$I - M^{-1}J = (I - T_2 J)(I - T_1 J), \tag{5.2.2}$$

one can generally expect M to be a good preconditioner if T_1 and T_2 complement each other by closely approximating J on different parts of the spectrum. Other ways of combining two or more preconditioners to solve a single linear system can be found in [15].

5.2.1 True-IMPES reduction

The CPR preconditioner, which operates on the matrix J of size $2N \times 2N$, also has the form (5.2.1):

$$M_{CPR}^{-1} = M_2^{-1}\big[I - JC(W^T JC)^{-1}W^T\big] + C(W^T JC)^{-1}W^T, \tag{5.2.3}$$

where W^T and C, of size $N \times 2N$ and $2N \times N$ respectively, are the restriction and prolongation operators; M_2, of size $2N \times 2N$, is typically a local preconditioner such as ILU. The goal of the first stage preconditioner is to form a pressure equation

$$A_p \delta p = -r_p,$$

where $A_p = W^T JC$, that can be solved easily and gives a meaningful approximate pressure solution δp. Different choices of W^T and C give rise to different first-stage preconditioners, which is the subject of study in [44]. One popular choice of the first-stage preconditioner, called the *True-IMPES* reduction, uses the IMPES pressure matrix directly; in this case, A_p is an elliptic operator, so efficient solvers such as algebraic multigrid [74] can be used to solve the pressure equation. In addition, since A_p is simply the pressure matrix that arises from a different time discretization, the solution δp is also a meaningful approximation of the FIM pressure solution, at least when Δt is small.

In the general black-oil case with n_p phases, it is possible to obtain the IMPES pressure matrix by manipulating J directly. We describe the procedure here informally (but see [44] for a detailed discussion). Since IMPES treats the transmissibility derivatives explicitly, one needs to first eliminate these terms from J. This can be done by performing a column sum (i.e., for each phase p, sum the equations corresponding to phase p over the whole domain): since mass is conserved, all the flux terms must cancel, so the transmissibility derivatives will also cancel out. Only accumulation terms remain, which means that

$$\hat{J}_{ss} := \text{Colsum}(J_{ss}) \quad \text{and} \quad \hat{J}_{ps} := \text{Colsum}(J_{ps})$$

are now diagonal matrices. Finally, the pressure equation is obtained by eliminating the pressure variables, which is equivalent to forming the Schur complement

$$A_p = J_{pp} - \hat{J}_{ps}\hat{J}_{ss}^{-1}J_{sp}.$$

The resulting pressure matrix A_p will have the same sparsity pattern as J_{pp} and J_{sp}, since the scaling matrix $\hat{J}_{ps}\hat{J}_{ss}^{-1}$ is diagonal and does not modify the sparsity pattern of J_{sp}. In the incompressible two-phase flow case, Lemma 5.1 shows that

$$\hat{J}_{ss} = -\hat{J}_{ps} = \frac{1}{\Delta t}D,$$

so we have the very simple relation $A_p = J_{ps} + J_{pp} = J_{tp}$. Thus, the restriction and prolongation operators are

$$W^T = \begin{bmatrix} I & I \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ I \end{bmatrix},$$

and the first-stage preconditioner becomes

$$T_1 = C(W^TJC)^{-1}W^T = \begin{bmatrix} 0 & 0 \\ J_{tp}^{-1} & J_{tp}^{-1} \end{bmatrix}.$$

We wish to investigate the effect of the CPR preconditioner on J by computing

$$M_{CPR}^{-1}J = M_2^{-1}J(I - T_1J) + T_1J.$$

A straightforward calculation shows that

$$J(I - T_1J) = \begin{bmatrix} J_{ss} - J_{sp}J_{tp}^{-1}J_{ts} & 0 \\ J_{ps} - J_{pp}J_{tp}^{-1}J_{ts} & 0 \end{bmatrix}.$$

Recall that $S_1 = J_{ss} - J_{sp}J_{tp}^{-1}J_{ts}$ is the Schur complement with respect to the (1,1)-block. If we partition M_2 into $M_2 = \begin{bmatrix} \tilde{J}_{ss} & \tilde{J}_{sp} \\ \tilde{J}_{ps} & \tilde{J}_{pp} \end{bmatrix}$ and define $\tilde{S}_1 = \tilde{J}_{ss} - \tilde{J}_{sp}\tilde{J}_{tp}^{-1}\tilde{J}_{ts}$, we obtain

$$M_{CPR}^{-1}J = \begin{bmatrix} \tilde{S}_1^{-1}S_1 & 0 \\ J_{tp}^{-1}J_{ts} - \tilde{J}_{tp}^{-1}\tilde{J}_{ts}\tilde{S}_1^{-1}S_1 & I \end{bmatrix}. \tag{5.2.4}$$

As a result, $M_{CPR}^{-1}J$ has $\lambda = 1$ as an eigenvalue with (geometric) multiplicity at least N, so the first-stage preconditioner clusters all the eigenvalues associated with the pressure part into the point $z = 1$. Equation (5.2.4) also implies that GMRES converges in at most $N + 1$ iterations in exact arithmetic. To see this, consider any matrix of the form

$$G := \begin{bmatrix} S & 0 \\ Y & I \end{bmatrix}.$$

Let $q(t) = \sum_{i=0}^{k} \beta_i t^i$ be the minimal polynomial of S, where $\beta_0 \neq 0$ if and only if S is nonsingular. Then since

$$G^{i+1} - G^i = \begin{bmatrix} S^{i+1} - S^i & 0 \\ YS^i & 0 \end{bmatrix},$$

we see that

$$\sum_{i=0}^{k} \beta_i (G^{i+1} - G^i) = 0.$$

So the minimal polynomial of G, $\tilde{q}(t)$, has degree at most $k + 1$, and $\tilde{q}(0) \neq 0$ if and only if S is nonsingular. In the case of $G = M_{CPR}^{-1}J$, $\tilde{q}(t)$ has degree at most $N + 1$; this implies the convergence of GMRES within $N + 1$ iterations, since the m-th residual r^m of GMRES satisfies

$$\|r^m\|_2 = \min_{\substack{p_m \in \mathbb{P}_m \\ p_m(0)=1}} \|p_m(G)r^0\|_2,$$

where $\mathbb{P}_m$ denotes the set of polynomials with degree at most m.

Given the role the matrix $\tilde{S}_1^{-1}S_1$ plays, the convergence behavior of CPR is predominantly dictated by how well the second-stage preconditioner M_2 approximates the Schur complement S_1 with respect to the transport problem.

5.2.2 Improved second-stage preconditioner via ordering

The choice of second-stage preconditioners has a significant impact on the effectiveness of the overall CPR preconditioner. Based on (5.2.4), it is clear that an effective second-stage preconditioner must perform well on the transport problem. Popular choices for the second-stage preconditioners include ILU(k) (typically $k = 0$) as well as block ILU(k), where the n_p-by-n_p blocks correspond to the n_p equations in n_p unknowns aligned with a given control volume. Even though both pointwise and cell-based block ILU(k) have similar performance in practice, the block variant is generally more robust and easier to analyze, since no special procedure is needed to handle accidental zero entries arising from residual saturations. For instance, the (i, j) entry in J_{sp} is generally nonzero if i and j are adjacent gridblocks, but can become zero occasionally if $\lambda_w = 0$ at the $i - j$ interface. When this happens, pointwise ILU will drop any fill-in that occurs at the (i, j) position, whereas block ILU will retain the fill-in entry. For the remainder of this section, we mainly focus on block ILU to avoid complications of this sort.

The effectiveness of $BILU(0)$ on the transport problem is demonstrated next.

Proposition 5.4. *Let A be the Jacobian (in block form) of a 1D flow problem, with the cells ordered from left to right. Then if the $BILU(0)$ factorization of A exists (i.e., if no singular diagonal block occurs during factorization), it is exact.*

Proof. Since A is block tridiagonal, no fill-in occurs during block Gaussian elimination, so the block LU and BILU(0) factorizations coincide. As a result, the BILU(0) factorization is exact. □

Note that ILU is only exact if the cell-based block form of the Jacobian is used. Fill-in necessarily occurs if the partitioned form of the Jacobian is used, since the Schur complement

$$S_2 := J_{pp} - J_{ps} J_{ss}^{-1} J_{sp}$$

is not tridiagonal. In fact, $J_{ps} J_{ss}^{-1}$ is in general a full lower-triangular matrix, which means S_2 is in general a full lower Hessenberg matrix. Thus, one should expect BILU(0) to be a better second-stage preconditioner than ILU on the partitioned matrix J.

It is usually difficult to ascertain *a priori* that the BILU(0) factorization exists for the general two-phase flow problem. However, in the special case of cocurrent flow, we can prove the existence of BILU(0) when a cell-based potential ordering is used.

Theorem 5.5. *Let J be the Jacobian corresponding to a cocurrent flow problem linearized at (S^ℓ, P^ℓ), and suppose the pressure profile P^ℓ satisfies the maximum principle. Assume the cell-centered grid admits a two-coloring. Let $A = PJP^T$ be the block form of the Jacobian, in which the cells are arranged in decreasing order of pressure. Then the block ILU(0) factorization of A exists with nonsingular factors L and U. Moreover, we have*

$$P^T(LU)P = J + E, \tag{5.2.5}$$

where $E = \begin{bmatrix} 0 & E_{sp} \\ 0 & E_{pp} \end{bmatrix}$.

In other words, BILU(0) is exact on the saturation part. The assumption that P^ℓ satisfies the maximum principle implies that the cell(s) with the lowest pressure must be on a Dirichlet boundary. Also note that this theorem is applicable to cocurrent flow problems in any dimension, and not just for 1D flows, as long as the grid is two-colorable. This applies to many grids of practical interest (Cartesian and other orthogonal grids, radial grids, etc.). In light of (5.2.2) and the fact that T_1 is exact on the pressure part, Theorem 5.5 indicates that BILU(0) should be an excellent preconditioner as long as E_{sp} and E_{pp} are not too large.

Proof. Let $A = PJP^T$ and A_{ij} be the 2×2 blocks. Let $A^{(k)}$ be the block $(N-k+1) \times (N-k+1)$ matrix that remains to be factored at the kth step, i.e., we have

$A = A^{(1)}$,

$$A^{(k)} = \begin{bmatrix} A^{(k)}_{kk} & A^{(k)}_{k,k+1} & \cdots & A^{(k)}_{kN} \\ A^{(k)}_{k+1,k} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A^{(k)}_{Nk} & \cdots & \cdots & A^{(k)}_{NN} \end{bmatrix},$$

$$A^{(k+1)}_{ij} = \begin{cases} 0, & A^{(k)}_{ij} = 0; \\ A^{(k)}_{ij} - A^{(k)}_{ik}(A^{(k)}_{kk})^{-1}A^{(k)}_{kj}, & A^{(k)}_{ij} \neq 0 \end{cases}$$

for $i, j \geq k+1$.

1. We argue that
$$A_{ij} = A^{(1)}_{ij} = \cdots = A^{(K)}_{ij} \tag{5.2.6}$$
whenever $i \neq j$ and $K \leq \min(i,j)$. This is true because a two-coloring exists for any Cartesian grid, i.e., one can partition the gridblocks $V = \{1, 2, \ldots, N\}$ into disjoint sets V_R and V_B such that $A_{ij} = 0$ whenever $i \neq j$ and either $i, j \in V_R$ or $i, j \in V_B$. Thus, when $i \neq j$, either $A^{(k)}_{ij} = 0$ or $A^{(k)}_{ik}(A^{(k)}_{kk})^{-1}A^{(k)}_{kj} = 0$, which implies (5.2.6). Thus, the only blocks that change during the elimination are the diagonal blocks $A^{(k)}_{ii}$.

2. Because of upstream weighting in the finite-volume discretization, we see that for $i < j$,
$$A_{ij} = \begin{bmatrix} 0 & X_{ij} \\ 0 & Y_{ij} \end{bmatrix}.$$
Thus, $A^{(k)}_{ik}(A^{(k)}_{kk})^{-1}A^{(k)}_{kj}$ also has the form
$$\begin{bmatrix} 0 & * \\ 0 & * \end{bmatrix}, \tag{5.2.7}$$
which means only the second column of $A^{(k)}_{ii}$ gets updated during the elimination.

So we have

$$A_{ii}^{(k)} = \begin{bmatrix} a_{ii} & X_{ii}^{(k)} \\ -b_{ii} & Y_{ii}^{(k)} \end{bmatrix}; \quad A_{ij}^{(k)} = A_{ij} = \begin{bmatrix} -a_{ij} & -X_{ij} \\ b_{ij} & -Y_{ij} \end{bmatrix} \quad (i \neq j). \tag{5.2.8}$$

3. Let $\gamma_i = \phi_i V_i / \Delta t \geq \gamma_{\min} > 0$. The following properties hold for $A = A^{(1)}$:

   - $a_{ij}, b_{ij}, X_{ij}, Y_{ij}$ are all non-negative;
   - $X_{ij} = X_{ji}$ and $Y_{ij} = Y_{ji}$ for all $i \neq j$,
   - $a_{jj} \geq \gamma_j + \sum_{i>j} a_{ij}$;
   - $b_{jj} \geq \gamma_j + \sum_{i>j} b_{ij}$;
   - $X_{ii} \geq \sum_{j \neq i} X_{ij}$;
   - $Y_{ii} \geq \sum_{j \neq i} Y_{ij}$;
   - For a cell i on the Dirichlet boundary, $X_{ii} + Y_{ii} > \sum_{j \neq i}(X_{ij} + Y_{ij})$.

We prove inductively that for any given k and any $i, j \geq k$,

(a) $X_{ii}^{(k)} \geq \sum_{j \geq k, j \neq i} X_{ij} \geq 0$;

(b) $Y_{ii}^{(k)} \geq \sum_{j \geq k, j \neq i} Y_{ij} \geq 0$.

(c) $X_{ii}^{(k)} + Y_{ii}^{(k)} > \sum_{j \geq k, j \neq i}(X_{ij} + Y_{ij}) \geq 0$ for a cell i on the Dirichlet boundary.

Then (a)–(c) together would imply that $A_{ii}^{(k)}$ is nonsingular for all $k \leq i$. Assume first that cell i is not on the Dirichlet boundary. Then by the maximum principle, there is at least one cell downstream from i. Thus,

$$\det A_{ii}^{(k)} = a_{ii} Y_{ii}^{(k)} + b_{ii} X_{ii}^{(k)} \geq \gamma_{\min}(X_{ii}^{(k)} + Y_{ii}^{(k)}) \geq \gamma_{\min} \lambda_{t,\min} > 0.$$

Similarly, if i is on the Dirichlet boundary, then (c) implies

$$\det A_{ii}^{(k)} \geq \gamma_{\min}(X_{ii}^{(k)} + Y_{ii}^{(k)}) > 0.$$

Clearly, conditions (a)–(c) are satisfied for $k = 1$. For the inductive step, we compute

$$\begin{aligned} A_{ik}^{(k)}(A_{kk}^{(k)})^{-1}A_{ki}^{(k)} &= \frac{1}{\det A_{kk}^{(k)}} \begin{bmatrix} -a_{ik} & -X_{ik} \\ b_{ik} & -Y_{ik} \end{bmatrix} \begin{bmatrix} Y_{kk}^{(k)} & -X_{kk}^{(k)} \\ b_{kk} & a_{kk} \end{bmatrix} \begin{bmatrix} 0 & -X_{ki} \\ 0 & -Y_{ki} \end{bmatrix} \\ &= \begin{bmatrix} 0 & dX_{ii}^{(k)} \\ 0 & dY_{ii}^{(k)} \end{bmatrix}. \end{aligned}$$

We have

$$\begin{aligned} \big(\det A_{kk}^{(k)}\big) dX_{ii}^{(k)} &= a_{ik}Y_{kk}^{(k)}X_{ki} - a_{ik}X_{kk}^{(k)}Y_{ki} + X_{ik}b_{kk}X_{ki} + X_{ik}a_{kk}Y_{ki} \\ &\le a_{ik}Y_{kk}^{(k)}X_{ki} - a_{ik}X_{ki}Y_{ki} + X_{ik}b_{kk}X_{ki} + X_{ik}a_{kk}Y_{ki} \\ &= X_{ik}\Big(a_{ik}Y_{kk}^{(k)} - a_{ik}Y_{ki} + b_{kk}X_{ki} + a_{kk}Y_{ki}\Big) \\ &= X_{ik}\Big(a_{ik}Y_{kk}^{(k)} + (a_{kk} - a_{ik})Y_{ki} + b_{kk}X_{ki}\Big) \\ &\le X_{ik}\Big(a_{ik}Y_{kk}^{(k)} + (a_{kk} - a_{ik})Y_{kk}^{(k)} + b_{kk}X_{kk}^{(k)}\Big) \\ &= X_{ik}(a_{kk}Y_{kk}^{(k)} + b_{kk}X_{kk}^{(k)}) \\ &= \big(\det A_{kk}^{(k)}\big)X_{ik}. \end{aligned}$$

Thus, $dX_{ii}^{(k)} \le X_{ik}$, and by a similar calculation, we get $dY_{ii}^{(k)} \le Y_{ik}$. Hence,

$$X_{ii}^{(k+1)} = X_{ii}^{(k)} - dX_{ii}^{(k)} \ge \Big(\sum_{\substack{j \ge k \\ j \ne i}} X_{ij}\Big) - X_{ik} \ge \sum_{\substack{j \ge k+1 \\ j \ne i}} X_{ij},$$

proving (a); (b) and (c) are proved similarly.

4. The above argument shows that the block ILU(0) factorization of A exists and has the form

$$L = \begin{bmatrix} I & & & \\ L_{21} & I & & \\ \vdots & & \ddots & \\ L_{N1} & \cdots & L_{N,N-1} & I \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1N} \\ & U_{22} & & \vdots \\ & & \ddots & U_{N-1,N} \\ & & & U_{NN} \end{bmatrix},$$

where

$$L_{ij} = \begin{cases} A_{ij}\big(A_{jj}^{(j)}\big)^{-1}, & i > j,\ A_{ij} \neq 0, \\ I, & i = j, \\ 0, & \text{otherwise}; \end{cases} \tag{5.2.9}$$

$$U_{ij} = \begin{cases} A_{ij}, & i < j,\ A_{ij} \neq 0, \\ A_{ii}^{(i)}, & i = j, \\ 0, & \text{otherwise}. \end{cases} \tag{5.2.10}$$

Clearly, L and U are both nonsingular. Each 2×2 block in the factorization error PEP^T has the form $A_{ik}A_{kk}^{(k)}A_{kj}$, which has the pattern shown in (5.2.7). Thus, after permutation, we get

$$E = \begin{bmatrix} 0 & E_{sp} \\ 0 & E_{pp} \end{bmatrix},$$

as required. □

As we pointed out in section 3.2.4, it is not necessary to perform an exact sorting on the cell pressures in order to obtain a potential ordering. Instead, a topological sort, which can be calculated in $O(N)$ time, suffices. This implies there exist many ways to order the cells in such a way that J_{ss} and J_{ps} are triangular. Although it is conceivable that the different topological orderings will lead to different ILU prconditioners, the next theorem shows that they are in fact identical up to permutation.

Theorem 5.6. *Assume the hypotheses of Theorem 5.5. Let $G = (V, E)$ be the upstream graph, i.e., V is the set of cells in the domain, and $(i, j) \in E$ iff (1) i is adjacent to j, and (2) either $P_i^\ell > P_j^\ell$ or $P_i^\ell = P_j^\ell$ and $i > j$. Let*

$$\sigma_1 : V \to \{1, \ldots, N\}$$
$$\sigma_2 : V \to \{1, \ldots, N\}$$

be two topological orderings of G, and define $A_r = \Pi_r A \Pi_r^T$ $(r = 1, 2)$, where the Π_r

are block $N \times N$ permutation matrices with

$$(\Pi_r)_{ij} = \begin{cases} I, & j = \sigma_r(i) \\ 0 & \textit{otherwise.} \end{cases}$$

Then

$$\Pi_1^T L_1 U_1 \Pi_1 = \Pi_2^T L_2 U_2 \Pi_2,$$

where L_r and U_r are the block ILU(0) factors of A_r.

Proof. Let $\tau_r : \{1, \ldots, N\} \to V$ be the inverse of σ_r, $r = 1, 2$. Based on the expressions for L_r and U_r given by (5.2.9) and (5.2.10), it suffices to show that the diagonal blocks of $\Pi_1^T U_1 \Pi_1$ and $\Pi_2^T U_2 \Pi_2$ are identical. These diagonal blocks are given by

$$(U_r)_{ii} = (A_r)_{ii}^{(i)} = (A_r)_{ii} - \sum_{k<i} (A_r)_{ik}((A_r)_{ii}^{(i)})^{-1}(A_r)_{ki},$$

but since $(A_r)_{ik} = 0$ unless $(\tau_r(k), \tau_r(i)) \in E$, we really have

$$(U_r)_{ii} = (A_r)_{ii} - \sum_{(\tau_r(k),\tau_r(i)) \in E} (A_r)_{ik}(U_r)_{kk}^{-1}(A_r)_{ki}.$$

Thus, for any $j \in V$, we have $(U_1)_{\sigma_1(j),\sigma_1(j)} = (U_2)_{\sigma_2(j),\sigma_2(j)}$ if and only if

$$(U_1)_{\sigma_1(k),\sigma_1(k)} = (U_2)_{\sigma_2(k),\sigma_2(k)} \qquad \text{for all } k \text{ such that } (k, j) \in E,$$

which is true by induction (recall that G is a directed acyclic graph). □

Theorem 5.6 says that there is essentially only one BILU(0) preconditioner that respects flow directions.

Structure of the factorization error

It is possible to describe the nonzero pattern of the error matrices E_{sp} and E_{pp} in terms of the upstream graph G. A fill-in entry is created (and subsequently dropped by BILU(0)) at position (i, j), with $i \neq j$, if there exists $k < i, j$ such that both A_{ik}

and A_{jk} is nonzero. In other words, if E_{sp} and E_{pp} are nonzero at position (i, j), then nodes i and j must be *siblings* in the upstream graph G, i.e., i and j must share the same parent k. This immediately provides an upper bound on the number of entries in E_{sp} and E_{pp}: the number of error entries due to the elimination of node k is given by $d_k(d_k - 1)$, where d_k is the out-degree of node k (i.e., the number of edges coming out of k). So the total number of entries in E_{sp} and E_{pp} is bounded by

$$\sum_i d_i(d_i - 1) = \sum_i d_i^2 - \sum_i d_i = |V|D^2 - |E|,$$

where D is the maximum out-degree of any node in G. On a Cartesian grid, the parameters are

- 2D problems: $D \leq 4$, $|E| \approx 5|V|$;

- 3D problems: $D \leq 6$, $|E| \approx 7|V|$.

So in either case, the error matrices are sparse, since the number of entries scales linearly with $|V|$. Moreover, the value of the entries are given by

$$\begin{bmatrix} 0 & (E_{sp})_{ij} \\ 0 & (E_{pp})_{ij} \end{bmatrix} = -A_{ik}(A_{kk}^{(k)})^{-1}A_{kj}.$$

Physically, this corresponds to the flux from cell j to cell i (traveling via k) that is generated by a change in pressure p_j. Since the potential ordering always orders the cells according to the major flow direction, the fluxes between siblings are generally much smaller than fluxes along upstream edges. This implies the error matrices E_{sp} and E_{pp} are small. Contrast this with a lexicographical ordering, where there is no guarantee that the flux between siblings should be small. Thus, a second-stage preconditioner that uses potential ordering should be more effective than one that uses the natural ordering. This is what we observe in our numerical examples (section 5.2.4).

5.2.3 Spectrum of the preconditioned matrix

To understand the effectiveness of two-stage CPR preconditioning, it is instructive to examine the spectrum of the preconditioned matrix $M_{CPR}^{-1}J$ and compare it with the spectrum obtained from other preconditioners. Generally speaking, iterative solvers such as GMRES perform well when the spectrum of $M^{-1}J$ consists of a few compact, well-separated clusters far away from the origin ($z = 0$ on the complex plane), and preferably close to $z = 1$. It is important to note that when $M^{-1}J$ is non-normal, its eigenvalues do not completely determine the convergence behavior of GMRES (cf. [40, 55]); however, spectral plots still have heuristic value, because they allow us to compare visually the quality of various preconditioners.

We have already seen that, thanks to the first stage exact pressure solve, the preconditioned matrix has N eigenvalues at $z = 1$, while the remaining eigenvalues are given by the spectrum of $\tilde{S}_1^{-1}S_1$. We compute these eigenvalues for the BILU(0) case. We have

$$\begin{aligned}
I - M_{CPR}^{-1}J &= (I - M_2^{-1}J)(I - C(W^TJC)^{-1}W^TJ) \\
&= M_2^{-1}(M_2 - J)(I - C(W^TJC)^{-1}W^TJ) \\
&= M_2^{-1}\begin{bmatrix} 0 & E_{sp} \\ 0 & E_{pp} \end{bmatrix}\begin{bmatrix} I & 0 \\ -J_{tp}^{-1}J_{ts} & 0 \end{bmatrix} \\
&= \begin{bmatrix} -\hat{E}_{sp}J_{tp}^{-1}J_{ts} & 0 \\ -\hat{E}_{pp}J_{tp}^{-1}J_{ts} & 0 \end{bmatrix}.
\end{aligned}$$

So $\tilde{S}_1^{-1}S_1 = I + \hat{E}_{sp}J_{tp}^{-1}J_{ts}$, where

$$\begin{aligned}
\begin{bmatrix} \hat{E}_{sp} \\ \hat{E}_{pp} \end{bmatrix} &= M_2^{-1}\begin{bmatrix} E_{sp} \\ E_{pp} \end{bmatrix} \\
&= \begin{bmatrix} \tilde{S}_1^{-1}\tilde{J}_{pp}\tilde{J}_{tp}^{-1} & -\tilde{S}_1^{-1}\tilde{J}_{sp}\tilde{J}_{tp}^{-1} \\ -\tilde{S}_2^{-1}J_{ps}J_{ss}^{-1} & \tilde{S}_2^{-1} \end{bmatrix}\begin{bmatrix} E_{sp} \\ E_{pp} \end{bmatrix} \\
&= \begin{bmatrix} \tilde{S}_1^{-1}(\tilde{J}_{pp}\tilde{J}_{tp}^{-1}E_{sp} - \tilde{J}_{sp}\tilde{J}_{tp}^{-1}E_{pp}) \\ \tilde{S}_2^{-1}(E_{pp} - J_{ps}J_{ss}^{-1}E_{sp}) \end{bmatrix}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\hat{E}_{sp}J_{tp}^{-1}J_{ts} &= \tilde{S}_1^{-1}\left[\tilde{J}_{pp}\tilde{J}_{tp}^{-1}(\tilde{J}_{sp} - J_{sp}) - \tilde{J}_{sp}\tilde{J}_{tp}^{-1}(\tilde{J}_{pp} - J_{pp})\right]J_{tp}^{-1}J_{ts} \\
&= \tilde{S}_1^{-1}\left[(\tilde{J}_{sp} - J_{sp}) - \tilde{J}_{sp}\tilde{J}_{tp}^{-1}(\tilde{J}_{tp} - J_{tp})\right]J_{tp}^{-1}J_{ts} \\
&= \tilde{S}_1^{-1}\left[\tilde{J}_{sp}\tilde{J}_{tp}^{-1} - J_{sp}J_{tp}^{-1}\right]J_{ts}.
\end{aligned}$$

It is interesting to compare the above expressions with the spectrum one would get with a single-stage BILU(0) preconditioner (i.e., no pressure solve). In that case, we would have

$$M_2^{-1}J = \begin{bmatrix} I & -\hat{E}_{sp} \\ 0 & I - \hat{E}_{pp} \end{bmatrix}.$$

Therefore, in the single-stage BILU case, we would still have termination within $N+1$ steps when flow is cocurrent, but the convergence behavior from iteration 1 to N is dictated by $I - \hat{E}_{pp}$, instead of $I + \hat{E}_{sp}J_{tp}^{-1}J_{ts}$. We expect the CPR preconditioner to outperform single-stage BILU(0) based on the following (somewhat heuristic) reasons:

1. If $\|J_{ts}\|$ is small (e.g., when the overall flow (total mass balance equations) is slowly varying with respect to the time-step size), then $\tilde{S}_1^{-1}S_1$ will be close to the identity matrix, whereas this is not the case for single-stage BILU. In many practical applications, the total velocity, which dictates J_{tp}, does not vary much within a time step, so CPR would have a significant advantage over BILU(0).

2. It can be shown (see Appendix E) that both $J_{sp}J_{tp}^{-1}$ and $J_{pp}J_{tp}^{-1}$ are similar to a symmetric positive semi-definite matrix, with eigenvalues between 0 and 1. Thus, even though factorization errors are present, one can also expect $\tilde{J}_{sp}\tilde{J}_{tp}^{-1}$, and hence the term $\tilde{J}_{sp}\tilde{J}_{tp}^{-1} - J_{sp}J_{tp}^{-1}$, to be relatively benign. On the other hand, one cannot bound the eigenvalues of $J_{ps}J_{ss}^{-1}$: when Δt is large, $J_{ps}J_{ss}^{-1}$ can have both very large eigenvalues (when $|\lambda_{o,i}| \gg |\lambda_{w,i}|$) and very small ones (when $|\lambda_{o,i}| \ll |\lambda_{w,i}|$). So any bound on $\hat{E}_{sp}$ is likely to be much tighter than a bound on $\hat{E}_{pp}$.

3. It is easy to see that $I - \hat{E}_{pp} = \tilde{S}_2^{-1}S_2$, so the use of single-stage BILU(0) is equivalent to preconditioning the Schur complement with respect to pressure

> (S_2) by a fixed-pattern ILU preconditioner. Contrast this with CPR, which attempts to precondition S_1, the Schur complement with respect to saturation. The two Schur complements S_1 and S_2 are related by
>
> $$\begin{aligned} S_1 &= J_{ss} - J_{sp}J_{tp}^{-1}J_{ts} = J_{ss}(I - J_{ss}^{-1}J_{sp}J_{tp}^{-1}J_{ts}), \\ S_2 &= J_{tp} - J_{ts}J_{ss}^{-1}J_{sp} = J_{tp}(I - J_{tp}^{-1}J_{ts}J_{ss}^{-1}J_{sp}). \end{aligned}$$
>
> Since the two matrices inside the parentheses have the same eigenvalues, it is evident that S_1 behaves more like the transport part J_{ss}, whereas S_2 behaves more like the elliptic part J_{tp}. In particular, one expects $\kappa(S_1)$ to scale like $O(\Delta t/h)$, whereas $\kappa(S_2)$ would be $O(1/h^2)$. We also know that fixed-pattern ILU preconditioners tend to perform poorly on elliptic problems. This indicates CPR should, in general, outperform single-stage BILU(0).

To illustrate these arguments, we show the spectral plots of J, $M_2^{-1}J$ and $M_{CPR}^{-1}J$, as well as their condition numbers, for various time-step sizes in Figures 5.1 and 5.2. For this test case, we have a 2D homogeneous reservoir (with uniform porosity), discretized on a 20 × 10 grid. A constant injection rate is imposed along the left edge, and pressure is held constant along the right edge, with no flow boundaries along the top and bottom. We see that the spectrum of J changes significantly as Δt varies. The condition number is very large for all cases, and there is no obvious clustering of eigenvalues, which means GMRES will likely perform poorly without preconditioning. When BILU(0) is used, the spectrum lies almost completely on the positive real axis, but the distribution is continuous and no obvious clustering exists; in fact, the spectrum looks very similar to one belonging to an elliptic operator (possibly due to J_{tp} appearing as a multiplicative factor in S_2). When two-stage CPR is used, the clustering around $z = 1$ becomes very obvious, and the high quality of the clustering is remarkably consistent across time steps.

Figures 5.3 and 5.4 show the same spectral plots when countercurrent flow is present. The same comments concerning the spectra of J and $M_2^{-1}J$ apply, except that the condition numbers become much higher. As for the CPR-preconditioned matrix, we still see a very good clustering of eigenvalues around $z = 1$, but the cluster

Table 5.1: Convergence behavior for the block ILU(0) and CPR preconditioners. Each figure represents the average number of GMRES iterations per Newton step required for convergence.

		$\Delta t = 1.6$	$\Delta t = 3.1$	$\Delta t = 7.8$
Cocurrent	BILU(0)	23.0	22.7	23.0
	CPR	3.7	4.3	5.0
Countercurrent	BILU(0)	22.4	22.0	22.0
	CPR	5.4	6.0	7.0

is not as tight, and we start to see more spreading along the positive real axis. This is probably due to the fact that M_2 is no longer exact with respect to saturation, and this factorization error manifests itself as a spreading of the eigenvalues. Fortunately, the outlying eigenvalues are well separated from one another, so GMRES should have little problem eliminating the subspaces associated with them within a few iterations. Table 5.1 shows the linear iteration counts per Newton step for both the CPR and block ILU(0) preconditioners on the 20×10 grid. For both the cocurrent and countercurrent flow cases, it is evident that the higher quality clustering produced by CPR does, in fact, translate into much faster convergence compared with block ILU(0).

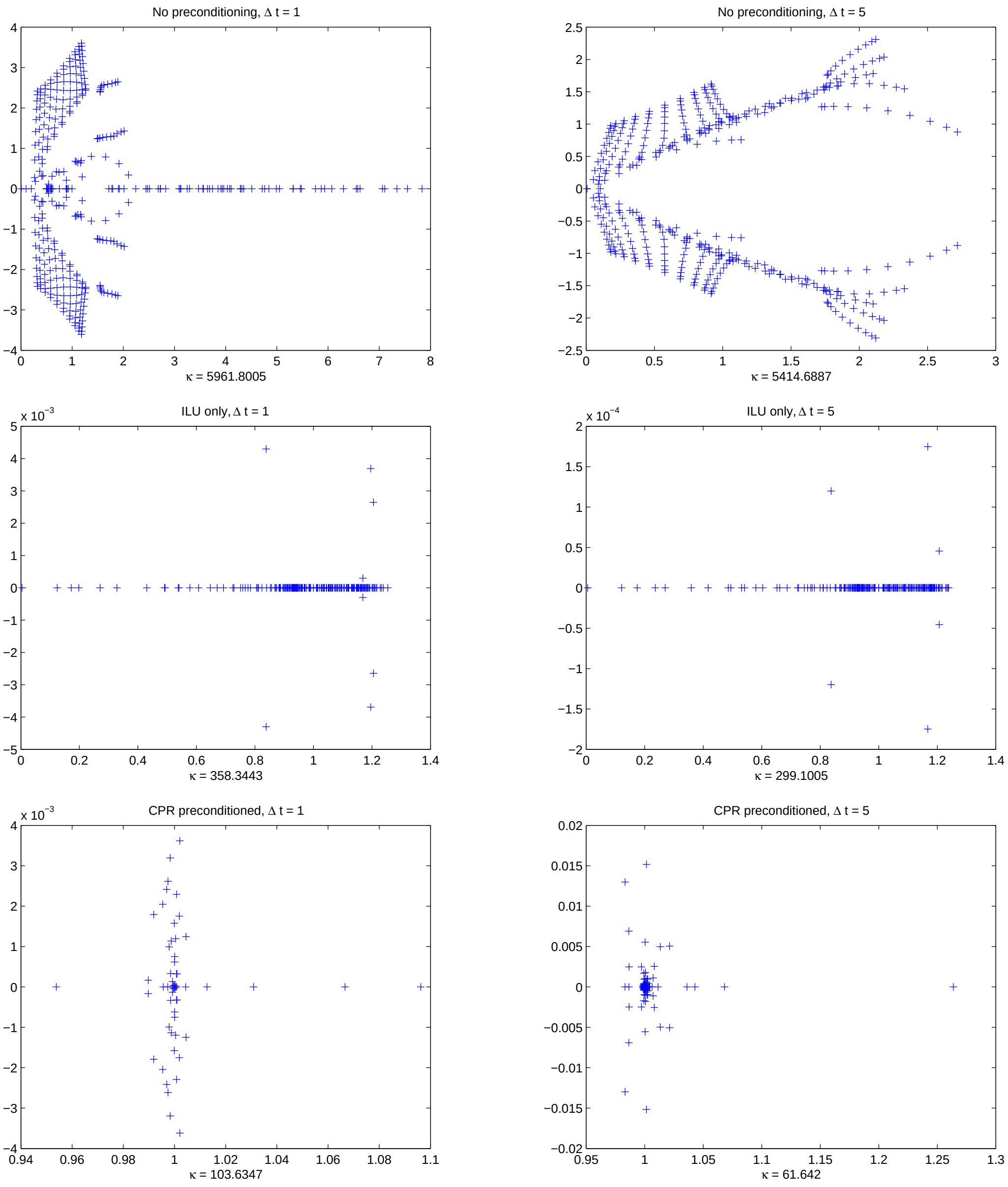


Figure 5.1: Spectra of Jacobian (no preconditioning), BILU(0) and CPR preconditioning for the cocurrent flow problem ($\Delta t = 1, 5$).

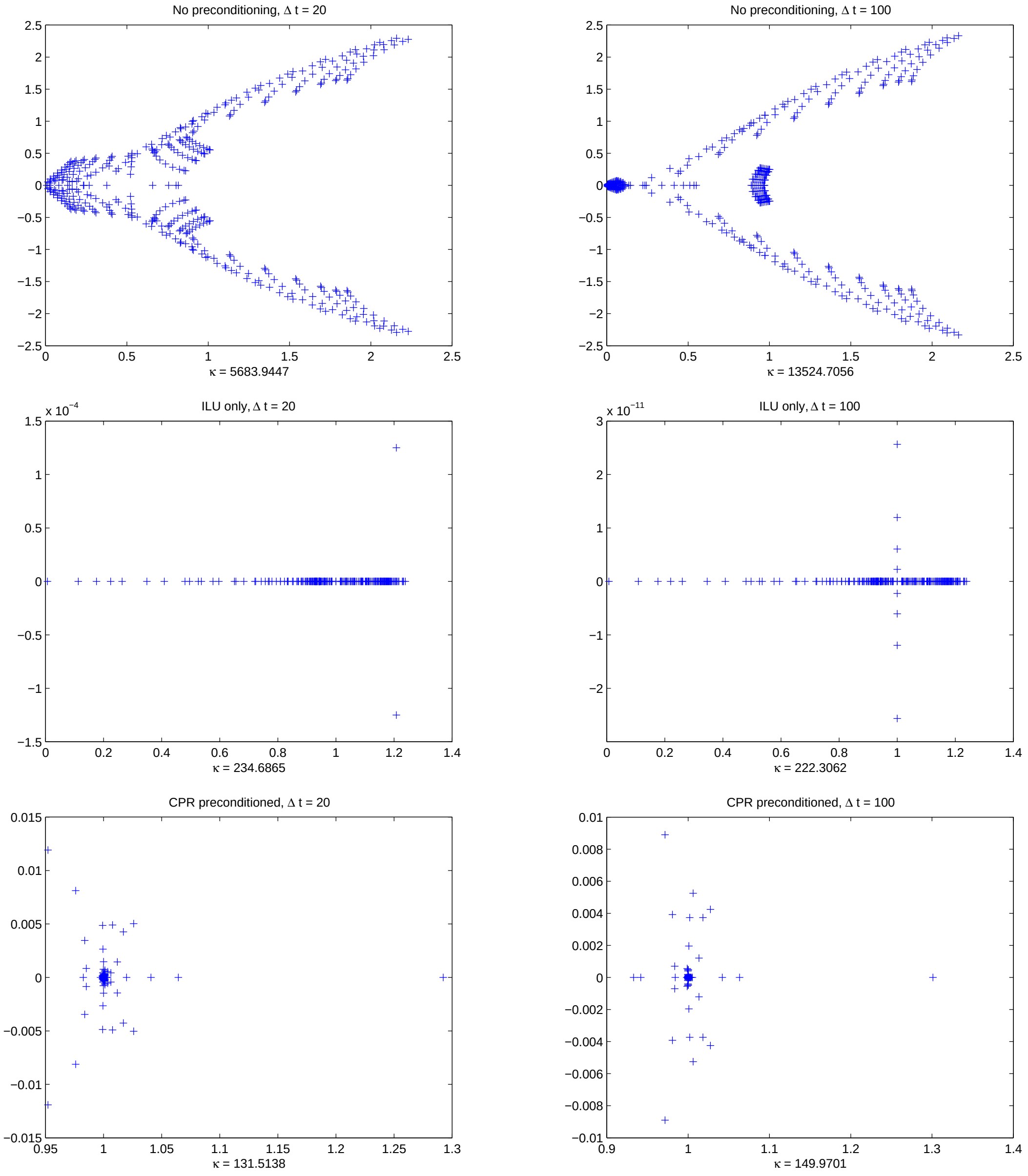


Figure 5.2: Spectra of Jacobian (no preconditioning), BILU(0) and CPR preconditioning for the cocurrent flow problem ($\Delta t = 20, 100$).

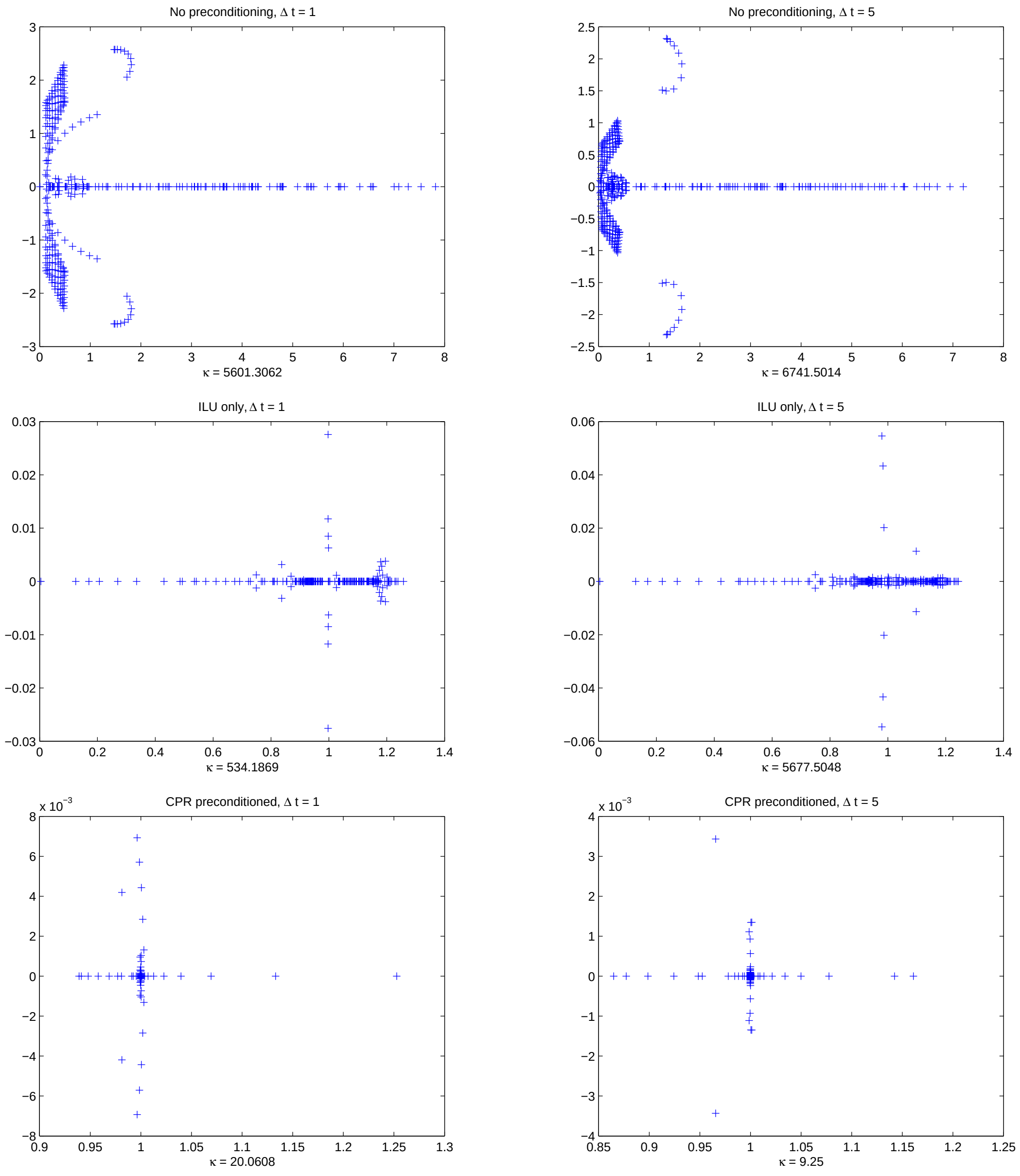


Figure 5.3: Spectra of Jacobian (no preconditioning), BILU(0) and CPR preconditioning for the countercurrent flow problem ($\Delta t = 1, 5$).

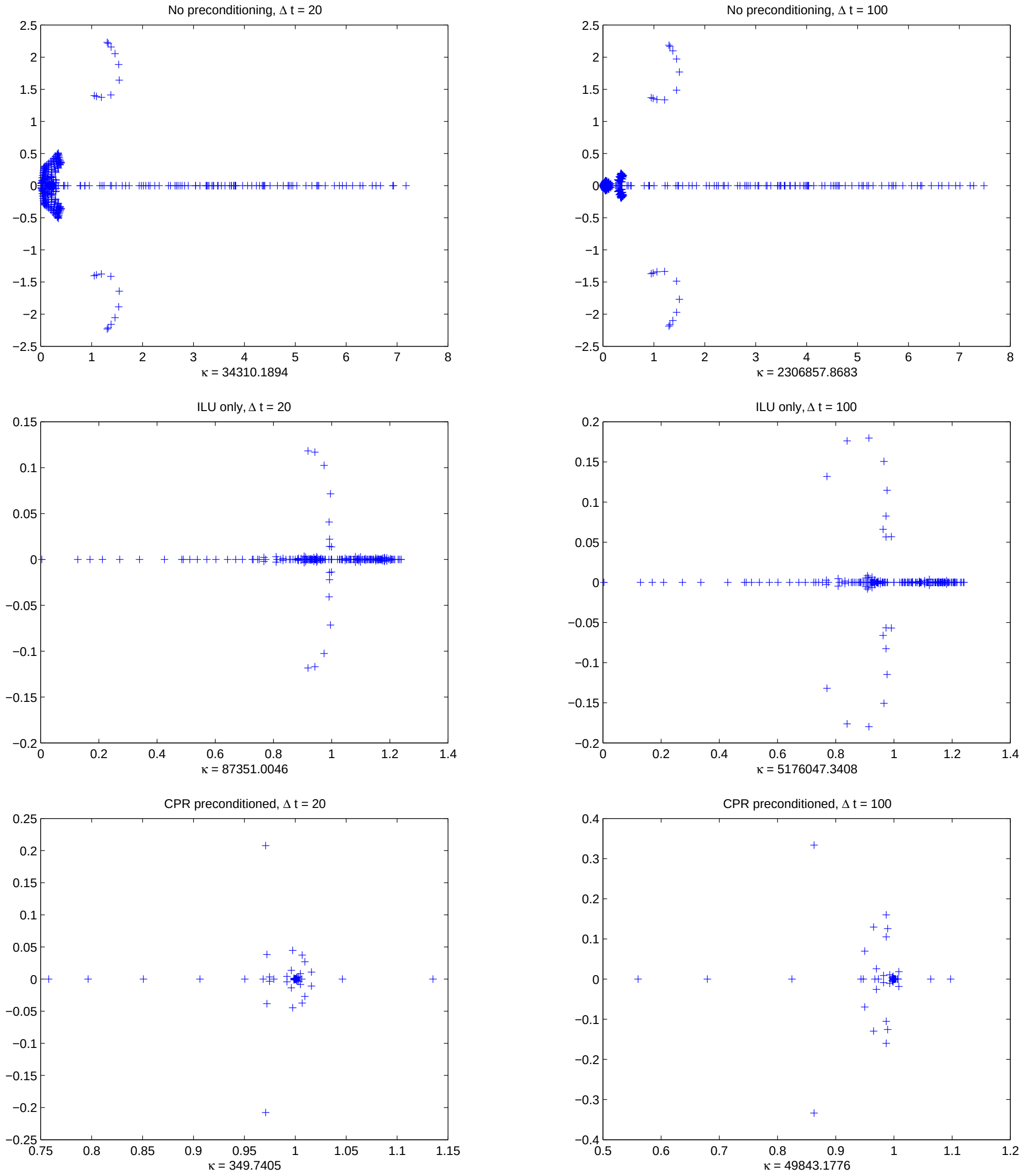


Figure 5.4: Spectra of Jacobian (no preconditioning), BILU(0) and CPR preconditioning for the countercurrent flow problem ($\Delta t = 20, 100$).

5.2.4 Numerical examples

To illustrate the importance of ordering on the CPR preconditioner, we provide the following two numerical examples. The first example shows how the performance of CPR with the standard ordering can vary significantly depending on the flow configuration, even on the same problem, whereas CPR with potential ordering is insensitive to flow configurations. The second example shows the effect of ordering on a 3D complex flow problem.

Quarter 5-spot problem

For this test problem, we have a two-dimensional reservoir that is discretized on a 20×20 grid. Water is injected through a well at one corner of the reservoir, at a constant rate of 0.005 pore volumes per day; a production well, maintained at a fixed pressure, is located at the opposite corner (see Figure 5.5). The no-flow condition is imposed on the remaining sections of the boundary. Quadratic relative permeabilities are used, with a mobility ratio of $M = 10$. The simulation is run until $T = 100$ days (0.5 pore volumes injected). The cells are numbered in lexicographical order (i.e., from left to right, then from bottom to top). We solve the same problem under two different configurations: in the first case (a), the injection and production wells are located at the lower-left and upper-right corners respectively, so that the lexicographical ordering coincides with the potential ordering. In the second case (b), the wells are located at the lower-right and upper-left corners instead, so the natural ordering is no longer a valid potential ordering. Table 5.2 shows the total iteration counts over the whole simulation, as well as running time information. (Data for single-stage BILU(0) are omitted, since there are many Newton steps within which BILU(0) fails to converge within 1500 iterations.) The two configurations require exactly the same number of time steps and Newton steps, as expected. However, the number of GMRES iterations for the two cases are significantly different (a 36% increase) when CPR-BILU(0) with lexicographical ordering is used. This difference is insignificant when potential ordering is used, which is expected since the upstream graphs for the two problems are isomorphic, i.e., the two graphs are the same up to relabeling. (The discrepancy in iteration counts is probably due to the inexact solve

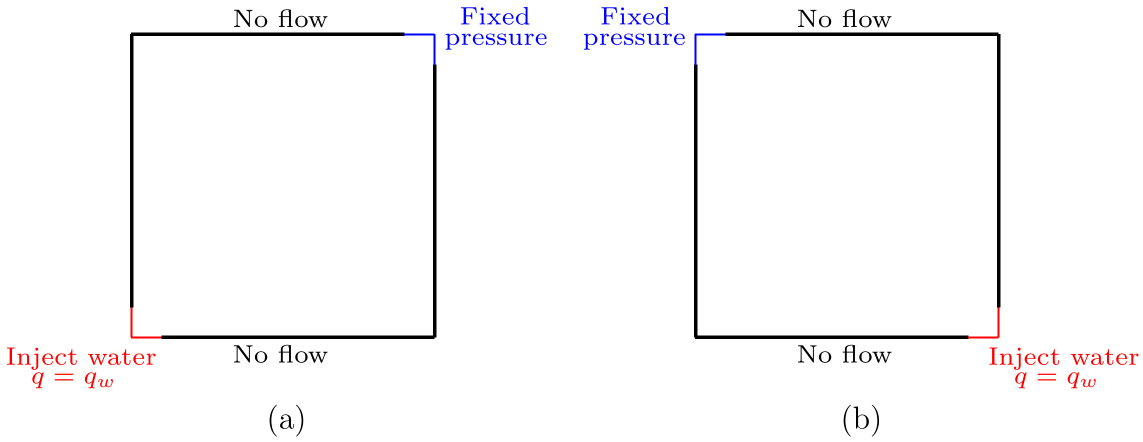


Figure 5.5: Two configurations of the quarter 5-spot problem.

in the first stage.) This example provides experimental confirmation of Theorem 5.6 and illustrates the ability of potential ordering to shield this type of grid orientation effect from the linear solver.

Upscaled SPE 10 problem

In this example, the reservoir is a 2 × 2 × 2 upscaling of the SPE 10 problem, i.e., it is identical to the one used in example 4.3.2 in Chapter 4. It is initially saturated with oil, and we inject water at the center of the reservoir at a rate of 0.0002 pore volumes per day (or 29 cell pore volumes per day). A production well, maintained at constant pressure, is completed at one corner of the reservoir. Once again, we test the solvers on two configurations:

(a) The production well is located at (1,1,:), so that the major direction of flow is aligned with the lexicographical ordering;

(b) The production well is located at (110,1,:), so that the major direction of flow is transverse to the lexicographical ordering.

We run the simulation to $T = 100$ days (0.02 pore volumes injected). Table 5.3 summarizes the runs for both preconditioners. The number of time steps and Newton steps are again exactly the same in all cases, indicating that the problems are

Table 5.2: Performance of CPR-ILU for the quarter 5-spot problem.

	Config. (a)		Config. (b)	
	Natural ordering	Potential ordering	Natural ordering	Potential ordering
No. of time steps	21	21	21	21
No. of Newton steps	80	80	80	80
No. of GMRES iterations	254	254	346	246
No. of AMG V-cycles	286	286	368	274
Total running time (sec)	1.37	1.45	1.49	1.43
– Top. sort (sec)	0	0.02	0	0.02
– Permutation (sec)	0	0.08	0	0.09
– BILU solve (sec)	0.10	0.14	0.15	0.09
– Pressure solve (sec)	0.32	0.27	0.39	0.33

completely equivalent. Even in configuration (a), the number of GMRES iterations decreases somewhat when potential ordering is used, because the lexicographical ordering is no longer a valid potential ordering because of the strong spatial heterogeneity of the permeability field. When configuration (b) is used instead, we observe an increase in GMRES iterations when the lexicographical ordering is used, since the major direction of flow is no longer aligned with this ordering. However, the iteration count is almost exactly the same when potential ordering is used, once again illustrating the invariance of potentially-ordered BILU(0) with respect to flow configuration details.

In our current implementation, the savings due to the use of potential ordering are rather modest, even though the GMRES iteration count decreases substantially. We believe this is due to our inefficient implementation. Currently, we physically permute the blocks of the Jacobian matrix into the potential ordering before feeding it into a library routine that computes the block ILU factorization. This simplifies the implementation, but adds unnecessary cost to the solver, because one can actually modify the ILU routine to use the unpermuted data structure, and only change the order of elimination when the factorization is computed. (This is what modern direct

Table 5.3: Performance of CPR-ILU for the upscaled SPE 10 problem.

	Config. (a)		Config. (b)	
	Natural ordering	Potential ordering	Natural ordering	Potential ordering
No. of time steps	37	37	37	37
No. of Newton steps	106	106	106	106
No. of GMRES iterations	389	349	447	351
No. of AMG V-cycles	524	502	570	504
Total running time (sec)	595.99	614.37	630.55	616.21
– Top. sort (sec)	0	4.49	0	4.57
– Permutation (sec)	0	30.79	0	30.93
– BILU solve (sec)	46.82	42.72	52.76	42.34
– Pressure solve (sec)	229.33	224.60	242.93	225.89

solvers typically do when a symmetric permutation is required.) Since the permutation step represents about 5% of the total running time, eliminating this step should lead to a significant performance improvement. Note that the cost of computing the topological ordering is insignificant, and it can be shared with other modules. For instance, if reduced Newton is used as a nonlinear solver, then a topological ordering would already have been calculated, so there would be no need to compute it again for the linear solver. If all these efficiency measures are taken, the potential-ordered CPR-ILU preconditioner should outperform lexicographical ordering in most practical cases.

5.3 Schur complement preconditioning

Recall that the first-stage True-IMPES preconditioning corresponds to the following choice of restriction and prolongation operators in the first stage:

$$W^T = \begin{bmatrix} I & I \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Another reasonable choice of W^T and C would be

$$W^T = \begin{bmatrix} -J_{ps}J_{ss}^{-1} & I \end{bmatrix}, \quad C = \begin{bmatrix} 0 \\ I \end{bmatrix},$$

which leads to the first stage preconditioner being

$$T_1 = C(W^T JC)^{-1}W^T = \begin{bmatrix} 0 & 0 \\ -S_2^{-1}J_{ps}J_{ss}^{-1} & S_2^{-1} \end{bmatrix}.$$

The overall preconditioned matrix $M^{-1}J$ then becomes

$$M^{-1}J = \begin{bmatrix} \tilde{S}_1^{-1}(J_{ss} - \tilde{J}_{sp}\tilde{J}_{tp}^{-1}J_{ps}) & 0 \\ 0 & I \end{bmatrix},$$

meaning that if M_2 is exact on saturation (e.g., BILU(0) for cocurrent flow), then the two-stage preconditioner would be exact. Note that the "pressure matrix" $W^T JC$ in this case would be

$$W^T JC = -J_{ps}J_{ss}^{-1}J_{sp} + J_{pp} = S_2,$$

meaning we are actually solving the Schur complement problem *with respect to pressure.* S_2 is in general a dense matrix; however, as we noted in section 4.1, we can perform the matrix-vector product $S_2 v$ with exactly the same computational cost as performing Jv, since multiplication with J_{ss}^{-1} is simply a forward substitution when we exploit potential ordering. It is thus worthwhile to attempt to devise effective preconditioners for the Schur complement problem. In fact, a good preconditioner for S_2 (i.e., one that converges as quickly as two-stage CPR on the full problem) would eliminate the need for a two-stage preconditioner on J, since one can always obtain the saturation solution from the pressure solution by a back substitution (i.e., a multiplication by J_{ss}^{-1}).

The structure of this section is as follows. First, we study the properties of S_2 by examining its spectrum, nonzero pattern and relative magnitudes of its entries. Then we look at different approximations to the Schur complement and how they behave as preconditioners for cocurrent and countercurrent cases.

5.3.1 Spectrum and nonzero pattern

We consider a pseudo-1D flow situation, where water is injected from one edge of the reservoir and pressure is held constant at the other edge. The reservoir is initially filled with water between the injection edge and the middle of the reservoir, while the remaining part is filled with oil. Flow is cocurrent (i.e. no gravity is present). For both the 1D (20 × 1) case and the 2D (20 × 20) case, we show the nonzero pattern of the full matrix in Figure 5.6. We also show the spectrum, as well as the absolute value of the entries, for the following time steps:

(a) 0.1 cell pore volumes,

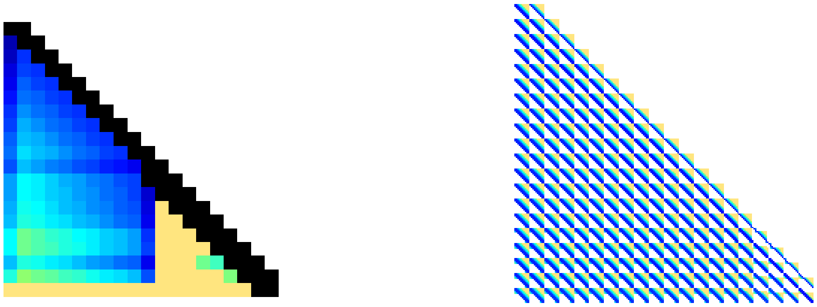
(b) 1 cell pore volume,

(c) 10 cell pore volumes.

The spectral and profile plots are shown in Figures 5.7 (for 1D problems) and 5.8. The profile plots show the magnitudes of the entries along a row of S_2 that corresponds to a gridblock near the center of the reservoir, behind the flood front.

Let us first look at the spectrum of S_2. When the time step is small, the transport problem contributes little to the spectrum of the Schur complement; the eigenvalue plot is similar to that of a positive definite elliptic operator. For the medium and large time step, though, we start to see a rather complicated spectral plot consisting of two parts: eigenvalues along the positive real axis corresponding to the pressure part, and complex conjugate pairs that arise from the saturation part of the problem. The plots for the 2D case are especially revealing, since the complex eigenvalues are roughly in the shape of a parabola and bear a striking resemblance to the pseudospectra of convection-diffusion operators [63]. Based on the spectral plots, we conclude that preconditioners that depend strongly on the matrix being nearly symmetric positive definite (such as algebraic multigrid) will probably perform poorly on problems with moderate to large time steps.

As for the nonzero pattern, darker colors in Figure 5.6 indicate larger magnitudes. In the 1D case, most of the energy (i.e., Frobenius norm) of the matrix lies within the tridiagonal part; even though the lower triangular part is technically nonzero, most



1D flow 2D flow

Figure 5.6: Nonzero pattern of S_2 for the 1D and 2D reservoirs. Darker colors indicate larger magnitudes.

of the entries outside the tridiagonal region are tiny and can be neglected. Thus, ILU(0) can potentially be a good preconditioner for the 1D Schur complement. In contrast, the 2D Jacobian has large entries outside the pentadiagonal region, and the magnitude of the fill-in entries increases as the time step is increased. Hence, it is unlikely that a preconditioner with small bandwidth (such as an ILU preconditioner induced from the partitioned matrix J) would be effective for 2D problems.

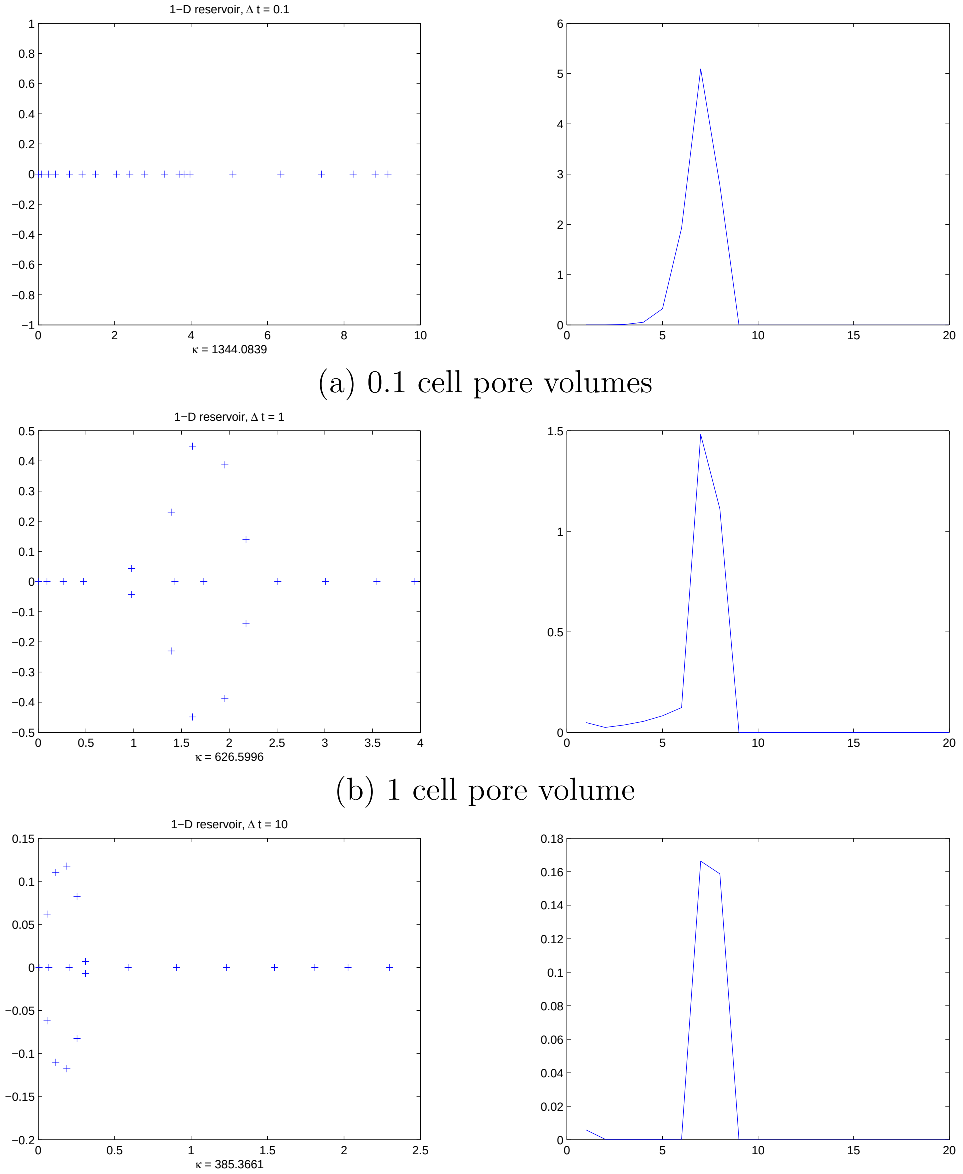
5.3.2 Convergence behavior

Recall the partitioned form J of the Jacobian matrix J:

$$J = \begin{bmatrix} J_{ss} & J_{sp} \\ J_{ps} & J_{pp} \end{bmatrix}.$$

We investigate the convergence behavior of the following preconditioners:

- $M_0 = J_{pp} - \mathrm{Colsum}(J_{ps})\,\mathrm{Colsum}(J_{ss})^{-1}J_{sp}$ (True-IMPES),
- $M_1 = J_{pp} - \mathrm{diag}(J_{ps})\,\mathrm{diag}(J_{ss})^{-1}J_{sp}$ (Quasi-IMPES),
- $M_2 = J_{pp} - J_{ps}\,\mathrm{diag}(J_{ss})^{-1}J_{sp}$,
- $M_3 = J_{pp} - \mathrm{diag}(J_{ps})\hat{J}_{ss}^{-1}J_{sp}$,

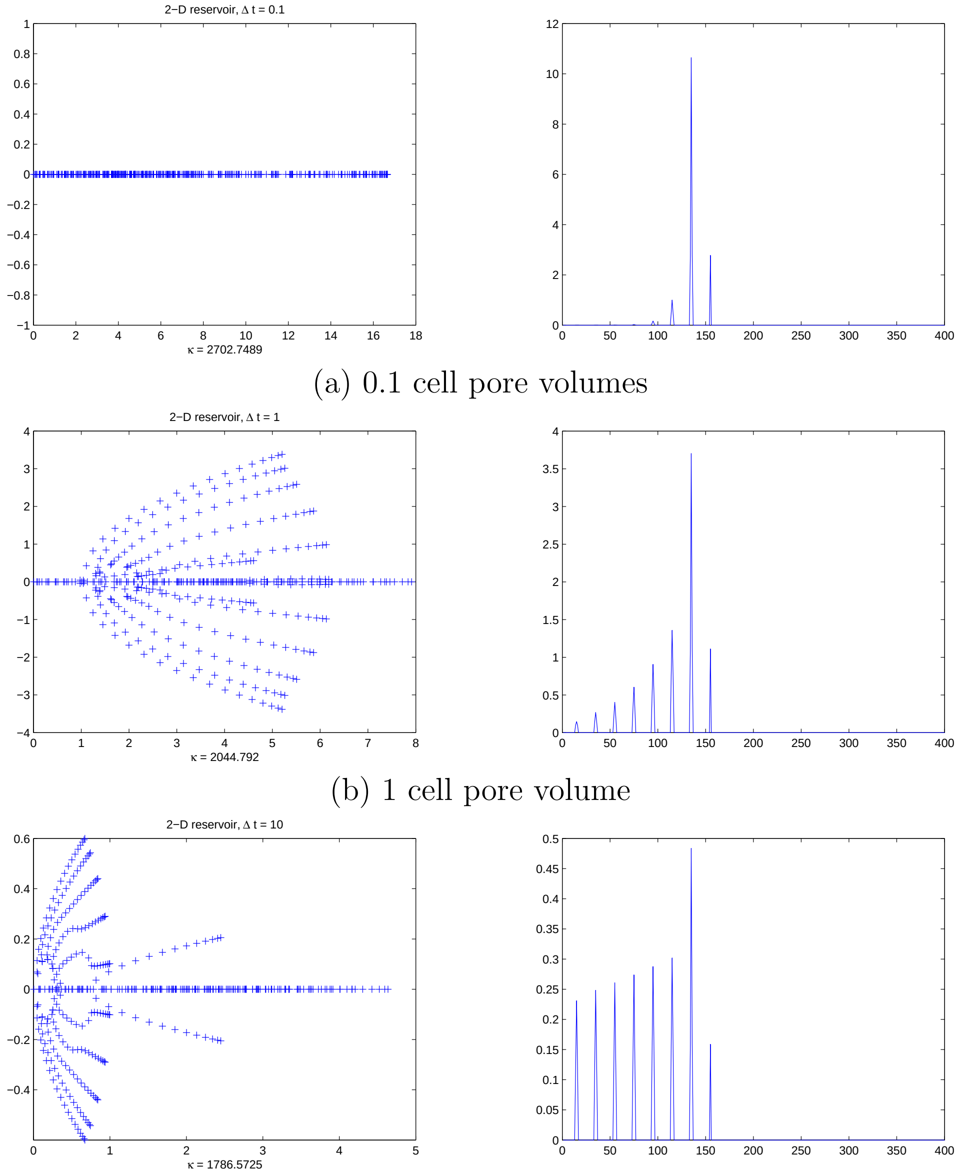


(a) 0.1 cell pore volumes

(b) 1 cell pore volume

(c) 10 cell pore volumes

Figure 5.7: Spectrum and nonzero profiles of S_2 for the 1D reservoir.



(a) 0.1 cell pore volumes

(b) 1 cell pore volume

(c) 10 cell pore volumes

Figure 5.8: Spectrum and nonzero profiles of S_2 for the 2D reservoir.

- $M_4 = J_{pp} - J_{ps}\hat{J}_{ss}^{-1}J_{sp}$,

where $\hat{J}_{ss}^{-1}$ is a first-order approximation of J_{ss}^{-1}, defined as follows. Suppose we order J_{ss} so that it is lower triangular. Then $J_{ss} = D - L = (I - LD^{-1})D$, where D is diagonal and L is strictly lower triangular. Then

$$\begin{aligned} J_{ss}^{-1} &= D^{-1}(I - LD^{-1})^{-1} \\ &= D^{-1}(I + LD^{-1} + \cdots + (LD^{-1})^{N-1}). \end{aligned}$$

Then the first order approximation is taken to be $\hat{J}_{ss}^{-1} = D^{-1}(D + L)D^{-1}$.

Tables 5.4 and 5.5 illustrate the rate of convergence for each of these preconditioners. The flow setting is the same as the 2D case in the previous section, except we now show results for both cocurrent and countercurrent flow. Each figure represents the average number of linear iterations per Newton iteration that GMRES takes to reduce the linear residual by a factor of 10^{-6}. The 'AMG' column corresponds to the case where the preconditioner is applied using one cycle of AMG, whereas the 'exact' column corresponds to the case where the preconditioner is applied using a direct method. The time step size Δt is measured in cell pore volumes injected. 'DNC' means the linear iteration does not converge within 100 iterations, and the approximate linear solution is so poor that it causes Newton's method to diverge.

For comparison purposes we include results for when the following preconditioners are used:

1. Induced ILU: single-stage preconditioner induced from the ILU(0) factorization of the partitioned matrix J.

2. AMG on S_2: the full Schur complement is handed to AMG, and one V-cycle is used per GMRES iteration;

3. CPR on J: two-stage preconditioner applied to the full Jacobian;

4. CPR on S_2: two-stage preconditioner induced from the CPR method.

The induced preconditioners (items (1) and (4)) are defined as follows. If M_J is a preconditioner for the full matrix J, then the induced preconditioner M_S is defined

Table 5.4: Convergence of GMRES in the absence of gravity.

	AMG			Exact		
Δt	1.6	3.1	7.8	1.6	3.1	7.8
M_0	10.7	19.3	24.7	4.0	8.3	10.0
M_1	10.0	13.3	15.3	3.7	6.0	7.0
M_2	11.7	17.7	21.7	4.3	8.3	11.0
M_3	11.0	29.0	45.0	4.3	7.3	9.0
M_4	21.3	33.3	41.0	3.0	5.0	5.0
Induced ILU				33.7	38.3	40.7
AMG on S_2	7.7	16.3	14.3			
CPR on S_2	5.7	8.7	11.3			
CPR on J	4.0	6.3	5.3			

as

$$M_S^{-1} = \begin{bmatrix} 0 & I \end{bmatrix} M_J^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix},$$

meaning the preconditioning step $z = M_S^{-1} r$ is computed via

$$z = \begin{bmatrix} 0 & I \end{bmatrix} M_J^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix}.$$

Note that the induced ILU preconditioner can always be applied exactly because if $M_A = L_A U_A$, then $M_S = L_S U_S$, where

$$L_S = R^T L_A R, \quad U_S = R^T U_A R, \quad R^T = \begin{bmatrix} 0 & I \end{bmatrix}$$

(see Chapter 14 in [68] for a proof).

Based on the convergence data, the following observations can be made:

1. The convergence rates of all the Schur complement methods have a fairly strong dependence on the time-step size. CPR on the full matrix, on the other hand, exhibits a convergence behavior that is nearly independent of Δt, which is consistent with the spectral plots of Figures 5.1–5.4.

Table 5.5: Convergence of GMRES in the presence of gravity.

	AMG			Exact		
Δt	1.6	3.1	7.8	1.6	3.1	7.8
M_0	19.5	42.3	63.2	8.5	16.8	22.6
M_1	16.8	27.5	86.2	6.8	12.0	32.2
M_2	14.2	24.5	55.2	7.3	12.3	22.8
M_3	17.5	45.0	>100	7.2	12.3	23.8
M_4	44.0	DNC	DNC	6.2	10.0	15.2
Induced ILU				35.0	42.0	63.3
AMG on S_2	18.0	DNC	DNC			
CPR on S_2	8.2	11.5	22.6			
CPR on J	5.7	6.8	6.6			

2. The “best” preconditioner depends on the flow situation. While Quasi-IMPES beats True-IMPES in the cocurrent case, the opposite is true for countercurrent flow.

3. A more accurate approximation of the Schur complement does not imply faster convergence when AMG is used. In particular, M_4 (which has the most fill among the M’s) and the exact Schur complement S_2 both do poorly in the countercurrent flow case when AMG is used. As expected, AMG has trouble when the matrix is far from being an elliptic operator. This is in contrast with the exact preconditioner case, where a more accurate preconditioner usually requires fewer iterations to converge.

4. Induced ILU performs poorly, since S_2 contains a large elliptic component.

5. Keeping off-diagonal blocks, or at least treating them properly, is important for convergence in the countercurrent flow case, as seen in the faster convergence of M_2 relative to the other preconditioners.

Given our comment about the performance of narrow-band preconditioners, it is not surprising that M_0 through M_4 have a hard time competing with the CPR preconditioner. What is surprising, though, is that the induced preconditioner M_S

requires more iterations to converge than M_{CPR}^{-1} on the full system, even though it is operating on a smaller system. One possible explanation is as follows. A direct (but tedious) calculation shows that

$$M_S^{-1} = (I + \tilde{S}_2^{-1}\tilde{J}_{ts}\tilde{J}_{ss}^{-1}J_{sp})J_{tp}^{-1},$$

where $\tilde{S}_2 = \tilde{J}_{pp} - \tilde{J}_{ps}\tilde{J}_{ss}^{-1}\tilde{J}_{sp}$ is the Schur complement of the second stage preconditioner M_2 with respect to pressure. If we assume cocurrent flow and that BILU(0) with potential ordering is used, we would have $\tilde{J}_{ss} = J_{ss}$ and $\tilde{J}_{ps} = J_{ps}$, so the preconditioned matrix $S_2M_S^{-1}$ would become

$$\begin{aligned} S_2M_S^{-1} &= (J_{tp} - J_{ts}J_{ss}^{-1}J_{sp} + S_2\tilde{S}_2^{-1}J_{ts}J_{ss}^{-1}J_{sp})J_{tp}^{-1} \\ &= I + (S_2\tilde{S}_2^{-1} - I)J_{ts}J_{ss}^{-1}J_{sp}J_{tp}^{-1}. \end{aligned}$$

Once again,the convergence behavior depends on how close $S_2\tilde{S}_2^{-1}$ is to the identity, when such a term is absent from $M_{CPR}^{-1}J$. This could explain why CPR on the Schur complement S_2 scales less well than CPR on the full Jacobian J.

Chapter 6

Conclusions

The efficient simulation of immiscible multiphase flow in porous media requires the use of nonlinear solvers and linear preconditioners that can take advantage of the underlying structure of the problem, such as flow direction information. The phase-based potential ordering in Chapter 3 exploits the upstream nature of the spatial discretization in order to triangularize the saturation part of the nonlinear system of equations. This ordering is valid for any flow configuration, and it can handle countercurrent flow due to gravity and capillarity. To compute the ordering, one simply needs to perform a topological sort on the upstream graph, so the time complexity scales linearly with the size of the grid. Moreover, this cost can be amortized over several Newton and time steps, since in practice flow directions reverse only sparingly.

The proposed phase-based potential ordering allows a partial decoupling of the transport problem from the flow problem, since the saturations can be computed via back substitution once the pressures are known. This allows us to derive a reduced-order Newton algorithm, which is the nonlinear analog of a Schur complement approach in matrix computations. We have proved that for 1D countercurrent flow, the reduced Newton method converges unconditionally for large Δt. In addition, a minor modification to the method (which can be thought of as pivoting) yields provable convergence for any time-step size. As demonstrated in various examples, reduced Newton has a much more robust convergence behavior than the usual Newton method, which translates into the ability to take larger time steps without risking divergence of the nonlinear iterations. This, in turn, leads to a more efficient and robust simulator overall.

Ordering techniques can also lead to improvements in the linear solver. For a cocurrent flow problem, a block ILU(0) factorization always exists provided the cells are ordered according to the phase potential, and this factorization is unique over all topological orderings. Moreover, this factorization is exact on the saturation part of the Jacobian. Since block ILU is used as the second stage of CPR preconditioning, exactness on saturation means that the pairing with True-IMPES reduction, which is exact on pressure, is practically ideal. Moreover, its uniqueness over topological orderings means CPR is much less sensitive to flow configuration variations if potential ordering is used. Spectral plots and numerical experiments demonstrate the power of this combination. Finally, experiments reveal that it is difficult to construct a preconditioner for the pressure Schur complement S_2 that rivals two-stage CPR in performance. This is likely because S_2 is a dense matrix that exhibits both advective and diffusive characters, as indicated by the spectral plots.

A rigorous analysis is performed on the phase-based upstream discretization. This discretization handles sonic points differently from the classical Godunov and Engquist-Osher schemes, since the upstream directions are obtained from the potential gradient of each phase, rather than by manipulating the fractional flow curve directly. Even though the numerical flux function becomes non-differentiable whenever the upstream direction changes, our analysis shows that the nonlinear algebraic system resulting from a fully-implicit time discretization has a unique bounded solution for any time-step size, and the resulting solution profiles are always monotonic. Since the analysis is based on the nonlinear Gauss-Seidel process, it also leads to an implementable algorithm for solving these nonlinear systems. The convergence rate is generally linear, but can become superlinear when the correct ordering is used. In addition, the phase-based upstream scheme satisfies an entropy inequality, so the method converges under mesh refinement. This is verified experimentally for a countercurrent flow problem. Finally, when a non-uniform grid is used, the solution accuracy is often comparable to the uniform-grid case, even though the maximum CFL number is usually much higher for the non-uniform grid. This reveals the real advantage of the fully-implicit method over an explicit scheme: the ability to handle the large CFL numbers that naturally arise from heterogeneity.

Future directions

In this section, we outline several possible future research directions stemming from our work.

Treatment of strong countercurrent flow due to gravity

In section 4.2.4, we showed that the reduced Newton method is expected to converge when the countercurrent flow due to gravity satisfies a backward CFL condition. On the other hand, when the backward CFL number is much larger than 1, it is possible for reduced Newton to cycle or diverge, especially when the initial guess is poor. In practical simulations, it is generally too restrictive to require a backward CFL number that is less than 1 everywhere. This is because the flow in regions far away from wells can be dominated by gravity segregation, since the total velocity is close to zero there. Thus, the backward CFL numbers in these regions (which are close to the foward CFL numbers) determine the convergence behavior of reduced Newton. Ongoing work focuses on hybridizing reduced Newton with a globally convergent scheme (e.g., nonlinear Gauss-Seidel) in such a way that reduced Newton handles regions with low backward CFL numbers, whereas regions with strong countercurrent flow are to be handled by the globally convergent scheme.

Extending reduced Newton to compositional models

Compositional simulations are even more expensive than black-oil simulations, especially when a large number of components are present. It would be beneficial to extend the ordering and reduction paradigm introduced in Chapters 3 and 4 to a compositional setting. A natural starting point would be the IMPSAT formulation [16], in which pressure and saturations are treated implicitly, whereas compositions (mole fractions of each component) are treated explicitly. Since compositions are not primary variables, the nonlinear algebraic system contains only pressure and saturations; in other words, the implicit part looks exactly like a black-oil system, so we can use reduced Newton without modifications. A possible difficulty is that in a compositional model, heavy hydrocarbon components are allowed to vaporize into the gas phase, whereas this is not allowed in the standard black-oil model. This could complicate the triangularization process, as both the oil and gas equations contain flow terms from both phases. However, since the amount of heavy components in the vapor phase is generally small for heavy oils, it may still be possible to triangularize the system by temporarily freezing or linearizing the S_g dependent terms in the oil equation. More numerical experiments and theory are needed to verify the effectiveness of this approach.

Extensions of stability analysis

The stability and convergence analysis presented in Chapter 2 are generally applicable to scalar hyperbolic conservation laws. This is adequate for one-dimensional flow, since the total velocity there is constant and known. However, for multiple dimensions, our existence and stability results apply only if we temporarily fix the total velocity field and solve the scalar transport problem on this frozen velocity field. Thus, our approach does not directly apply to the fully-implicit method, which solves for both the updated flow field and saturations in a coupled fashion. Our next step is to extend our analysis to handle this coupling properly. Ongoing work also focuses on extending the analysis to handle three-phase flow.

Appendix A

Pressure Equation Derivation

Here we derive the pressure equation (1.1.18). Assume no gravity, capillarity or source terms. Then the phase equations are given by:

$$\text{Water:} \qquad \frac{\partial}{\partial t}(\phi\rho_w S_w) - \nabla \cdot (K\lambda_w\rho_w\nabla p) = 0, \tag{A.1}$$

$$\text{Oil:} \qquad \frac{\partial}{\partial t}(\phi\rho_o S_o) - \nabla \cdot (K\lambda_o\rho_o\nabla p) = 0, \tag{A.2}$$

$$\text{Gas:} \qquad \frac{\partial}{\partial t}(\phi\rho_g S_g + \phi\rho_o R_s S_o) - \nabla \cdot (K\lambda_g\rho_g\nabla p + K\lambda_o\rho_o R_s\nabla p) = 0. \tag{A.3}$$

We assume ρ_w, ρ_o, ρ_g, ϕ and R_s are all smooth functions of pressure p, and p is differentiable with respect to t. We multiply the water equation by $1/\rho_w$ and expand the time derivative:

$$\frac{1}{\rho_w}(\phi'\rho_w + \phi\rho_w')\frac{\partial p}{\partial t}S_w + \phi\frac{\partial S_w}{\partial t} - \frac{1}{\rho_w}\nabla \cdot (K\lambda_w\rho_w\nabla p) = 0. \tag{A.4}$$

For the oil equation, we multiply by $(\rho_g - \rho_o R_s)/(\rho_o\rho_g)$:

$$\left(\frac{\rho_g - \rho_o R_s}{\rho_o\rho_g}\right)(\phi'\rho_o + \phi\rho_o')\frac{\partial p}{\partial t}S_o + \phi\left(1 - \frac{\rho_o R_s}{\rho_g}\right)\frac{\partial S_o}{\partial t} - \left(\frac{\rho_g - \rho_o R_s}{\rho_o\rho_g}\right)\nabla\cdot(K\lambda_o\rho_o\nabla p) = 0. \tag{A.5}$$

Finally, we multiply the gas equation by $1/\rho_g$:

$$\begin{aligned}\frac{1}{\rho_g}\Big[(\phi'\rho_g+\phi\rho_g')S_g+(\phi'\rho_oR_s+\phi\rho_o'Rs+\phi\rho_oR_s')S_o\Big]\frac{\partial p}{\partial t}\\ +\phi\frac{\partial S_g}{\partial t}+\phi\frac{\rho_oR_s}{\rho_g}\frac{\partial S_o}{\partial t}-\frac{1}{\rho_g}\nabla\cdot(K\lambda_g\rho_g\nabla p+K\lambda_o\rho_oR_s\nabla p)=0.\end{aligned} \tag{A.6}$$

We now add (A.4)–(A.6) together. First, the sum of the saturation derivatives is

$$\phi\frac{\partial S_w}{\partial t}+\phi\left(1-\frac{\rho_oR_s}{\rho_g}\right)\frac{\partial S_o}{\partial t}+\phi\frac{\partial S_g}{\partial t}+\phi\frac{\rho_oR_s}{\rho_g}\frac{\partial S_o}{\partial t}=\phi\left(\frac{\partial S_w}{\partial t}+\frac{\partial S_o}{\partial t}+\frac{\partial S_g}{\partial t}\right)=0,$$

since $S_w+S_o+S_g\equiv 1$. Next, the coefficient of $\partial p/\partial t$ is given by

$$\begin{aligned}&\phi'\Big[S_w+S_o\Big(1-\frac{\rho_oR_s}{\rho_g}\Big)+S_g+\frac{\rho_oR_s}{\rho_g}S_o\Big]\\ &\qquad+\phi\left[\frac{\rho_w'}{\rho_w}S_w+\frac{\rho_o'}{\rho_o}\Big(1-\frac{\rho_oR_s}{\rho_g}\Big)S_o+\frac{\rho_g'}{\rho_g}S_g+\Big(\frac{\rho_o'R_s}{\rho_g}+\frac{\rho_oR_s'}{\rho_g}\Big)S_o\right]\\ &=\phi'(S_w+S_o+S_g)+\phi\left[\frac{\rho_w'}{\rho_w}S_w+\left(\frac{\rho_o'}{\rho_o}+\frac{\rho_oR_s'}{\rho_g}\right)S_o+\frac{\rho_g'}{\rho_g}S_g\right]\\ &=\phi(c_r+c_w+c_o+c_g)=:\phi c_T,\end{aligned}$$

where

$$c_r=\frac{\phi'}{\phi},\qquad c_w=\frac{\rho_w'}{\rho_w},\qquad c_o=\frac{\rho_o'}{\rho_o}+\frac{\rho_oR_s'}{\rho_g},\qquad c_g=\frac{\rho_g'}{\rho_g}.$$

So the pressure equation is

$$\begin{aligned}\phi c_T\frac{\partial p}{\partial t}-\Bigg[\frac{1}{\rho_w}\nabla\cdot(K\lambda_w\rho_w\nabla p)+\left(\frac{\rho_g-\rho_oR_s}{\rho_o\rho_g}\right)\nabla\cdot(K\lambda_o\rho_o\nabla p)\\ +\frac{1}{\rho_g}\nabla\cdot(K\lambda_g\rho_g\nabla p+K\lambda_o\rho_oR_s\nabla p)\Bigg]=0.\end{aligned} \tag{A.7}$$

We can simplify (A.7) further by assuming that p is differentiable with respect to the spatial variable x. We use the identity

$$\nabla\cdot(f\mathbf{v})=\nabla f\cdot\mathbf{v}+f\nabla\cdot\mathbf{v},$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{v} : \mathbb{R}^n \to \mathbb{R}^n$ are differentiable functions. The water term can be written as

$$\begin{aligned}\frac{1}{\rho_w}\nabla \cdot (K\lambda_w \rho_w \nabla p) &= \frac{1}{\rho_w}\left[(\rho_w' \nabla p) \cdot (K\lambda_w \nabla p) + \rho_w \nabla \cdot (K\lambda_w \nabla p)\right] \\ &= K\lambda_w c_w |\nabla p|^2 + \nabla \cdot (K\lambda_w \nabla p).\end{aligned}$$

Similarly, the oil term becomes

$$\begin{aligned}\left(\frac{\rho_g - \rho_o R_s}{\rho_o \rho_g}\right) \nabla \cdot (K\lambda_o \rho_o \nabla p) &= \frac{1}{\rho_o}\left(1 - \frac{\rho_o R_s}{\rho_g}\right)\left[(\rho_o' \nabla p) \cdot (K\lambda_o \nabla p) + \rho_o \nabla \cdot (K\lambda_o \nabla p)\right] \\ &= \left(1 - \frac{\rho_o R_s}{\rho_g}\right)\left[K\lambda_o \frac{\rho_o'}{\rho_o} |\nabla p|^2 + \nabla \cdot (K\lambda_o \nabla p)\right].\end{aligned}$$

Finally, the gas term takes the form

$$\begin{aligned}&\frac{1}{\rho_g}\nabla \cdot (K\lambda_g \rho_g \nabla p + K\lambda_o \rho_o R_s \nabla\ p) \\ &\qquad = \frac{1}{\rho_g}\Big[(\rho_g' \nabla p) \cdot (K\lambda_g \nabla p) + \rho_g \nabla \cdot (K\lambda_g \nabla p) \\ &\qquad\qquad\qquad + (\rho_o' R_s + \rho_o R_s') \nabla p \cdot (K\lambda_o \nabla p) + \rho_o R_s \nabla \cdot (K\lambda_o \nabla p)\Big] \\ &\qquad = K\lambda_g c_g |\nabla p|^2 + \nabla \cdot (K\ \lambda_g \nabla p) + \frac{\rho_o R_s}{\rho_g} \nabla \cdot (K\lambda_o \nabla p) \\ &\qquad\qquad\qquad + \frac{\rho_o R_s}{\rho_g}\frac{\rho_o'}{\rho_o} K\lambda_o |\nabla p|^2 + \frac{\rho_o R_s'}{\rho_g} K\lambda_o |\nabla p|^2.\end{aligned}$$

Substituting the above terms into (A.7) gives

$$\phi c_T \frac{\partial p}{\partial t} - \nabla \cdot (K\lambda_T \nabla p) - \chi_T K |\nabla p|^2 = 0, \tag{A.8}$$

where $\lambda_T := \lambda_w + \lambda_o + \lambda_g$ is the total mobility and $\chi_T := \lambda_w c_w + \lambda_o c_o + \lambda_g c_g$ is the mobility-weighted compressibility.

Appendix B

Diagonal Dominance and L^1-Accretivity

Here we prove the equivalence between column diagonal dominance and m-accretivity in the L^1-norm for linear maps over $\mathbb{R}^n$. Recall that for the space $L^1(\mathbb{R}^n)$, A is m-accretive if it is continuous and for any $u, v \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} (A(u)_i - A(v)_i) \operatorname{sgn}(u_i - v_i) \geq 0. \tag{B.1}$$

Theorem B.1. *Let $A : \mathbb{R}^n \to \mathbb{R}^n$ be a linear map with matrix $A = [a_{ij}]$. Then A is m-accretive if and only if A is column diagonally dominant, i.e.,*

$$a_{jj} \geq \sum_{i \neq j} |a_{ij}| \qquad \textit{for } j = 1, \ldots, n. \tag{B.2}$$

Proof. Since A is linear, it suffices to show equivalence between condition (B.2) and

$$\sum_{i=1}^{n} (Au)_i \operatorname{sgn}(u_i) \geq 0 \tag{B.3}$$

for any $u \in \mathbb{R}^n$. Assume (B.3) holds for any vector u. For a given $\varepsilon > 0$, define

$$u^{(j)} = (-\varepsilon \operatorname{sgn}(a_{1j}), \ldots, -\varepsilon \operatorname{sgn}(a_{j-1,j}), 1, -\varepsilon \operatorname{sgn}(a_{j+1,j}), \ldots, -\varepsilon \operatorname{sgn}(a_{nj}))^T.$$

Then

$$Au^{(j)} = A_j + \varepsilon v^{(j)},$$

where A_j is the j-th column of A and $\|v^{(j)}\|_1 \leq n\|A\|_1$. Since $\operatorname{sgn}(u_i^{(j)}) = -\operatorname{sgn}(a_{ij})$ for $i \neq j$, we obtain

$$\sum_{i=1^n} (Au^{(j)})_i \operatorname{sgn}(u_i^{(j)}) = a_{jj} - \sum_{i \neq j} |a_{ij}| + \varepsilon \sum_{i=1}^{n} v_i^{(j)} \operatorname{sgn}(u_i^{(j)}),$$

which must be non-negative by (B.3). Thus, we have

$$a_{jj} - \sum_{i \neq j} |a_{ij}| \geq -\varepsilon \sum_{i=1}^{n} v_i^{(j)} \operatorname{sgn}(u_i^{(j)}) \geq -n\varepsilon \|A\|_1,$$

which is true for all j. Letting $\varepsilon \to 0$ yields column diagonal dominance, as required.

Conversely, assume A is column diagonally dominant. Then so is AD, where D is a diagonal matrix with $d_{ii} > 0$. Now, for any $u \in \mathbb{R}^n$,

$$\sum_{i=1}^{n} (Au)_i \operatorname{sgn}(u_i) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} u_j \operatorname{sgn}(u_i) = \sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij}|u_j|) \operatorname{sgn}(u_i) \operatorname{sgn}(u_j). \tag{B.4}$$

If u has no zero entries, the above is equivalent to evaluating $s^T M s$, where s is a vector of ± 1s, and $M = AU$, $U = \operatorname{diag}(|u_1|, \ldots, |u_n|) > 0$, so that M is also column diagonally dominant. Thus,

$$\begin{aligned} s^T M s &= \sum_{j=1}^{n} \sum_{i=1}^{n} m_{ij} s_i s_j \\ &= \sum_{j=1}^{n} \left[m_{jj} s_j^2 + \sum_{i \neq j} m_{ij} s_i s_j \right] \geq \sum_{j=1}^{n} \left[m_{jj} - \sum_{i \neq j} |m_{ij}| \right] \geq 0, \end{aligned}$$

so (B.3) holds for u. The general case where u has zero entries is similar, except the double summation will skip over any index i or j for which u_i or u_j is zero. □

Appendix C

Convergence of the Cascade Method

Consider a one-dimensional model problem with

- incompressible flow,
- an injection boundary condition on the left,
- a pressure boundary condition on the right, and
- no countercurrent flow (e.g. horizontal reservoir with no capillarity).

The continuous form of the problem is given by the conservation law

$$\phi(x)\frac{\partial S_p(x)}{\partial t} + \frac{\partial u_p(x)}{\partial x} = 0, \quad x_L < x < x_R \tag{C.1}$$

for $p = w, o$, with

$$u_p(x) = -K(x)k_{rp}(S_w(x))\frac{dp(x)}{dx}$$

and boundary conditions

$$\begin{aligned} u_p(x_L) &= q_{p,L}, \\ p(x_R) &= p_R. \end{aligned}$$

Proposition C.1. *For the above 1D model problem, the Appleyard-Cheshire Cascade method [4] converges in two iterations, provided the cells are ordered from upstream to downstream (left to right). In particular, the saturation of each cell will be correct at the end of the first iteration, and the pressures will be correct at the end of the second iteration.*

Proof. Under the Cascade (left-to-right) ordering, the discretized equations have the form

$$\begin{aligned}\frac{\phi_i(S_{w,i}-S_{w,i}^{old})}{\Delta t}+\frac{1}{\Delta x}\left(K_{li}k_{rw}(S_{w,i})\frac{p_i-p_{i+1}}{\Delta x}-FI_{w,i}\right)&=0,\\ \frac{\phi_i(S_{w,i}^{old}-S_{w,i})}{\Delta t}+\frac{1}{\Delta x}\left(K_{li}k_{ro}(S_{w,i})\frac{p_i-p_{i+1}}{\Delta x}-FI_{o,i}\right)&=0.\end{aligned} \tag{C.2}$$

Let the exact solution be $S_{w,i}^*$ and $p_{w,i}^*$, $i=1,\ldots,N$, and let the initial guess be $S_{w,i}^{(0)}$ and $p_{w,i}^{(0)}$. Consider the first iteration of the Cascade method. In line 3 in Figure 3.1, the pressures are updated to $p_i^{(1)}$, but this has no impact on convergence in this model problem. The saturations $S_{w,i}$ are updated inside the loop from lines 4 to 8. We show by induction that at the ith step of the loop, $S_{w,j}$ and $FO_{p,j}$ are correct for $j<i$.

For the base case, let $i=1$. The single-cell problem becomes

$$\begin{aligned}\frac{\phi_1(S_{w,1}-S_{w,1}^{old})}{\Delta t}+\frac{1}{\Delta x}\left(K_{12}k_{rw}(S_{w,1})\frac{p_1-p_2^{(1)}}{\Delta x}-Aq_{w,L}\right)&=0,\\ \frac{\phi_1(S_{w,1}^{old}-S_{w,1})}{\Delta t}+\frac{1}{\Delta x}\left(K_{12}k_{ro}(S_{w,1})\frac{p_1-p_2^{(1)}}{\Delta x}-Aq_{o,L}\right)&=0.\end{aligned} \tag{C.3}$$

which we can solve for $S_{w,1}$ and p_1. Since the exact solution also solves the single-cell problem, the uniqueness of solutions tells us that

$$S_{w,1}=S_{w,1}^* \quad \text{and} \quad p_1-p_2^{(1)}=p_1^*-p_2^*.$$

Thus, $S_{w,1}$ is exact, and the outward flux

$$\begin{aligned} FO_{p,1} &= K_{12} k_{rp}(S_{w,1})(p_1 - p_2^{(1)})/\Delta x \\ &= K_{12} k_{rp}(S^*_{w,1})(p^*_1 - p^*_2)/\Delta x \end{aligned}$$

is exact as well. This proves the base case. For $i > 1$, note that the outward fluxes for $j = 1, \ldots, i-1$ are assumed to be exact. This means the Cascade solution, and the exact solution at cell i, both solve the same single-cell problem. Hence, $S_{w,i} = S^*_{w,i}$, and the outward fluxes will match as well. Thus, the induction step goes through, and we have $S_{w,i} = S^*_{w,i}$ for all i after one iteration. It follows that during the second iteration of the Cascade method, in which we solve the linearized problem

$$J \begin{bmatrix} \delta S^{(2)} \\ \delta p^{(2)} \end{bmatrix} = -r^{(2)}, \tag{C.4}$$

we get $\delta S^{(2)} = 0$, which means (1) the transmissibility coefficients are exact, and (2) the fully implicit problem and the IMPES problem have the same pressure solution. But since the residual function is linear (affine) in pressure, solving (C.4) will yield the exact pressure, i.e.

$$p^*_i = p^{(1)} + \delta p^{(2)}.$$

So at the end of the second iteration, both the saturations and the pressures are correct, and the Cascade method converges to the solution. □

Appendix D

Nonsingularity of J_{ss}

Proposition D.1. *Let the relative permeability functions k_{rw} and k_{ro} be such that $dk_{rw}/dS_w \geq 0$ and $\partial k_{ro}/\partial S_o \geq 0$. Then $J_{ss} = \partial F_s/\partial S$ is nonsingular.*

Proof. Since J_{ss} is a lower triangular matrix, it suffices to show that none of its diagonal entries is zero. A typical oil conservation equation for cell i is

$$F_{oi} = \frac{\phi S_{oi}\rho_o(p_i)}{\Delta t} + \sum_{l \text{ adjacent to } i} K_{il} H_{o,il}(\Phi_{oi} - \Phi_{ol}) + F_{cap}, \tag{D.1}$$

where

$$H_{o,il} = \begin{cases} k_{ro}(S_i)\rho_o(p_i)/\mu_o(p_i) & \text{if } \Phi_{oi} \geq \Phi_{ol}, \\ k_{ro}(S_l)\rho_o(p_l)/\mu_o(p_l) & \text{if } \Phi_{oi} < \Phi_{ol}, \end{cases}$$

and F_{cap} denotes capillary forces, which are independent of S_o. Hence

$$\frac{\partial F_{oi}}{\partial S_{oi}} = \frac{\phi\rho_o(p_i)}{\Delta t} + K_{il}\frac{\partial H_{o,il}}{\partial S_{oi}}(\Phi_{oi} - \Phi_{ol}). \tag{D.2}$$

The accumulation term $\phi\rho_o(p_i)/\Delta t$ will always be positive. The sign of the flux term depends on the upstream direction. If $\Phi_{oi} \geq \Phi_{ol}$, then

$$\frac{\partial H_{o,il}}{\partial S_{oi}} = \frac{\rho_o(p_i)}{\mu_o(p_i)}\frac{\partial k_{ro}}{\partial S_o}(S_{oi}) \geq 0$$

by assumption. On the other hand, if $\Phi_{oi} < \Phi_{ol}$, then $H_{o,il}$ is independent of S_{oi}, so the derivative is zero. Thus, the flux derivative will always be non-negative, which means $\partial F_{oi}/\partial S_{oi} > 0$ for all cells i. The argument for the water equations is similar. Thus, J_{ss} has a positive diagonal, so it is nonsingular. □

Under certain mild conditions (to be specified below), the Stone I and II models (cf. [6]) can be shown to satisfy $\partial k_{ro}/\partial S_o \geq 0$, as required by Proposition D.1. Note that we are only concerned with saturations inside the region

$$D = \{(S_w, S_o, S_g) \,|\, S_w \geq S_{wc}, S_o \geq S_{om}, S_g \geq 0, S_w + S_o + S_g = 1\},$$

where S_{wc} is the connate water saturation and S_{om} is the minimum oil saturation at which oil is simultaneously displaced by water and gas. Also note that the derivative $\partial k_{ro}/\partial S_o$ is taken along the line $S_w =$ constant, so by the relation $S_w + S_o + S_g = 1$, the criterion $\partial k_{ro}/\partial S_o \geq 0$ is equivalent to $\partial k_{ro}/\partial S_g \leq 0$, which turns out to be more natural to show.

Proposition D.2. *Assume $dk_{rog}/dS_g \leq 0$. Then for saturations in D, the Stone I model satisfies $\partial k_{ro}/\partial S_g \leq 0$ provided $\partial S_{om}/\partial S_g \geq -\frac{1}{2}$.*

Proof. The Stone I model is defined as $k_{ro}(S_w, S_g) = k_{rocw} S_o^* \beta_w \beta_g$, where

$$\beta_w = \frac{k_{row}(S_w)/k_{rocw}}{1 - S_w^*}, \quad \beta_g = \frac{k_{rog}(S_g)/k_{rocw}}{1 - S_g^*}, \quad k_{rocw} = k_{row}|_{S_w = S_{wc}},$$

and the normalized saturations are defined as

$$S_w^* = \frac{S_w - S_{wc}}{1 - S_{wc} - S_{om}}, \quad S_o^* = \frac{S_o - S_{om}}{1 - S_{wc} - S_{om}}, \quad S_g^* = \frac{S_g}{1 - S_{wc} - S_{om}}.$$

Combining all these relations, we see that $k_{ro} = U(S_w, S_g, S_{om})/V(S_w, S_g, S_{om})$, where

$$U = (1 - S_w - S_{om} - S_g)(1 - S_{wc} - S_{om}) k_{row}(S_w) k_{rog}(S_g),$$
$$V = (1 - S_{wc} - S_{om} - S_g)(1 - S_w - S_{om}).$$

Remembering that $S_{om} = S_{om}(S_w, S_g)$, we deduce that

$$\begin{aligned}\frac{\partial k_{ro}}{\partial S_g} &= \frac{1}{V^2}\left[\left(V\frac{\partial U}{\partial S_g} - U\frac{\partial V}{\partial S_g}\right) + \frac{\partial S_{om}}{\partial S_g}\left(V\frac{\partial U}{\partial S_{om}} - U\frac{\partial V}{\partial S_{om}}\right)\right] \\ &= \frac{1}{V^2}\left[R_1 + R_2 \cdot \frac{\partial S_{om}}{\partial S_g}\right],\end{aligned}$$

so the sign of $\partial k_{ro}/\partial S_g$ is determined by the quantity within the square brackets. After some manipulation, we get

$$\begin{aligned}R_1 &= -(S_w - S_{wc})(1 - S_{wc} - S_{om})(1 - S_w - S_{om})k_{row}k_{rog} \\ &\quad + (1 - S_{wc} - S_{om} - S_g)(1 - S_w - S_{om})(1 - S_{wc} - S_{om})\times \\ &\quad\quad (1 - S_w - S_{om} - S_g)k_{row}k'_{rog} \\ &\le -(S_w - S_{wc})(1 - S_{wc} - S_{om})(1 - S_w - S_{om})k_{row}k_{rog} \\ &\le 0,\end{aligned}$$

since $k'_{org} \le 0$. In addition, we get

$$\begin{aligned}R_2 &= -S_g(S_w - S_{wc})\big[(1 - S_w - S_{om} - S_g) + (1 - S_{wc} - S_{om})\big]k_{row}k_{rog} \\ &\le 0.\end{aligned}$$

Hence $R_1 + R_2 \cdot \frac{\partial S_{om}}{\partial S_g} \le 0$ if either $\partial S_{om}/\partial S_g \ge 0$ or

$$\left|\frac{\partial S_{om}}{\partial S_g}\right| \le \frac{(S_w - S_{wc})(1 - S_{wc} - S_{om})(1 - S_w - S_{om})}{S_g(S_w - S_{wc})\big[(1 - S_w - S_{om} - S_g) + (1 - S_{wc} - S_{om})\big]}. \tag{D.3}$$

But since $S_g \le 1 - S_w - S_{om}$ and $1 - S_w - S_{om} - S_g \le 1 - S_{wc} - S_{om}$, we see that

$$\frac{(S_w - S_{wc})(1 - S_{wc} - S_{om})(1 - S_w - S_{om})}{S_g(S_w - S_{wc})\big[(1 - S_w - S_{om} - S_g) + (1 - S_{wc} - S_{om})\big]} \ge \frac{1}{2}.$$

Thus, in order to ensure that $\partial k_{ro}/\partial S_g \le 0$, it is sufficient to require either $\partial S_{om}/\partial S_g \ge 0$ or $|\partial S_{om}/\partial S_g| \le \frac{1}{2}$, which is equivalent to requiring $\partial S_{om}/\partial S_g \ge -\frac{1}{2}$. □

Note that if the Fayers and Matthews [35] model for S_{om} is used, we would have

$$\frac{\partial S_{om}}{\partial S_g} = -\frac{S_{orw} - S_{org}}{1 - S_{wc} - S_{org}},$$

so the condition in Proposition D.2 would be satisfied as long as $S_{orw} - S_{org}$ is small, which is usually the case. In particular, the monotonicity condition is always satisfied whenever $S_{orw} = S_{org}$.

Proposition D.3. *Assume that $dk_{rg}/dS_g \geq 0$, $dk_{rog}/dS_g \leq 0$, and that k_{rw} and k_{row} are convex functions of S_w. Then for all saturations in D, the Stone II model satisfies $\partial k_{ro}/\partial S_g \leq 0$.*

Proof. The Stone II model is defined as

$$k_{ro}(S_w, S_g) = k_{rocw}\left[\left(\frac{k_{row}}{k_{rocw}} + k_{rw}\right)\left(\frac{k_{rog}}{k_{rocw}} + k_{rg}\right) - (k_{rw} + k_{rg})\right].$$

Differentiating with respect to S_g gives

$$\frac{\partial k_{ro}}{\partial S_g} = \left(\frac{k_{row}}{k_{rocw}} + k_{rw} - 1\right)k'_{rg} + \left(\frac{k_{row}}{k_{rocw}} + k_{rw}\right)\frac{k'_{rog}}{k_{rocw}}.$$

The second term is clearly non-positive because $k'_{rog} \leq 0$. To show that the first term is also non-positive, first note that $k'_{rg} \geq 0$. Next, define $g(S_w) = k_{rw} + k_{row}/k_{rocw}$. Then $g(S_{wc}) = g(1 - S_{orw}) = 1$. But since g is convex, it must be that $g(S_w) \leq 1$ for all $S_{wc} \leq S_w \leq 1 - S_{orw}$. So $g(S_w) - 1 \leq 0$, which implies the first term is non-positive as well. Hence, we have shown that $\partial k_{ro}/\partial S_g \leq 0$, as required. □

Appendix E

Properties of Pressure Matrices

This appendix deals with the spectral properties of various combinations of the pressure matrices J_{sp}, J_{pp} and J_{tp}. These properties are useful in evaluating the relative importance of various terms that appear in the preconditioned matrices in Chapter 5.

Let $G = (V, E)$ be a connected undirected graph with nodes V and edges E. Suppose the nodes V can be partitioned into $V = V^{int} \cup V^{bdy}$, where $V^{bdy} \neq \emptyset$. (For our purposes, V^{int} consists of the control volumes in the domain; an edge in E is either an interface separating two cells, or the face of a boundary cell that is subject to a pressure boundary condition; V^{bdy} consists of “ghost cells” outside the domain that are used by the finite volume method to deal with pressure boundary conditions.) Suppose there exists a function $\sigma : E \to [0, \infty)$ that assigns a non-negative weight (transmissibility) to each edge in E, and let σ_{ij} denote the weight assigned to edge (i, j). Then we can define a $|V^{int}| \times |V^{int}|$ matrix M^σ by

$$M^\sigma_{ij} = \begin{cases} \sum_{(i,l)\in E} \sigma_{il} & i = j, \\ -\sigma_{ij} & i \neq j,\ (i,j) \in E, \\ 0 & i \neq j,\ (i,j) \notin E. \end{cases} \tag{E.1}$$

Then M^σ is a symmetric M-matrix, and by Gershgorin theorem its eigenvalues are non-negative, so that M^σ is positive semi-definite. If in addition $\sigma > 0$ then M^σ is irreducible, so by the Peron-Frobenius theorem it is also nonsingular, i.e. symmetric positive definite. Moreover, for any constant $c > 0$ we have $M^{c\sigma} = cM^{\sigma}$.

Given two weight functions σ and τ we say $\sigma \leq \tau$ if $\sigma_{ij} \leq \tau_{ij}$ for all edges (i, j). The following lemma is a slight modification of a theorem by Ostrowski and Reich (cf. [78]).

Lemma E.1. *Let $A = M - N$, where $A = A^*$, A and M are both nonsingular, and define $Q = M + M^* - A$. If A is positive definite and Q is positive semi-definite, then $\rho(M^{-1}N) \leq 1$, where $\rho(\cdot)$ is the spectral radius. In addition, if Q is positive definite, then $\rho(M^{-1}N) < 1$.*

Proof. Define $B = M^{-1}N = I - M^{-1}A$. It follows that if $Bu = \lambda u$, $u \neq 0$, then

$$Au = (1 - \lambda)Mu,$$

where $\lambda \neq 1$ since A is nonsingular. Taking the inner product of both sides with u yields

$$u^*Au = (1 - \lambda)u^*Mu,$$

but since A is symmetric positive definite, we also have

$$u^*Au = (1 - \bar{\lambda})u^*M^*u.$$

Adding these relaions yields

$$\begin{aligned} u^*(M + M^*)u &= \left(\frac{1}{1-\lambda} + \frac{1}{1-\bar{\lambda}}\right) u^*Au \\ &= 2\Re\left(\frac{1}{1-\lambda}\right) u^*Au, \end{aligned}$$

which can be rewritten as

$$\frac{u^*(Q + A)u}{u^*Au} = 1 + \frac{u^*Qu}{u^*Au} = 2\Re\left(\frac{1}{1-\lambda}\right).$$

Since A is positive definite and Q is positive semi-definite, we must have $2\Re\left(\frac{1}{1-\lambda}\right) \geq 1$, with strict inequality if Q is positive definite. If we write $\lambda = \alpha + i\beta$, it follows that

$$\frac{2(1-\alpha)}{(1-\alpha)^2 + \beta^2} \geq 1,$$

which yields $\alpha^2 + \beta^2 = |\lambda|^2 \leq 1$ (again with strict inequality if Q is positive definite). □

Corollary E.2. *Let σ and τ be weight functions on the edges E. If $\tau > 0$ and $0 \leq \sigma \leq \tau$, then $\rho((M^\tau)^{-1}M^\sigma) \leq 1$.*

Proof. Let $M = M^\tau$ and $N = -M^\sigma$ in Lemma E.1. Then $A = M^\tau + M^\sigma$ and $Q = M^\tau - M^\sigma$, which corresponds to matrices with weights $\tau + \sigma > 0$ and $\tau - \sigma \geq 0$, so that A is symmetric positive definite and Q is symmetric positive semi-definite. Thus, we have $\rho((M^\tau)^{-1}M^\sigma) \leq 1$ by Lemma E.1, as required. □

The above corollary immediately implies $\rho(J_{sp}J_{tp}^{-1}) \leq 1$ and $\rho(J_{pp}J_{tp}^{-1}) \leq 1$, since λ_w, λ_o are both bounded above by λ_T. The corollary also leads to a bound on the condition number of M^σ:

Theorem E.3. *Let σ and τ be weight functions on the edges E. If there exist constants $0 < b \leq B$ such that $0 < b\tau \leq \sigma \leq B\tau$, then*

$$\kappa_2(M^\sigma) \leq \frac{B}{b}\kappa_2(M^\tau),$$

where $\kappa_2(A) = \|A\|_2 \, \|A^{-1}\|_2$ is the 2-norm condition number.

Proof. Let $M^\tau = R^2$, where R is the symmetric square root of M^τ. In other words, $R = U\Lambda^{1/2}U^T$, where $M^\tau = U\Lambda U^T$ is the spectral decomposition of the symmetric positive definite matrix M^τ. Then by the above corollary we must have

$$\begin{aligned} b\rho(R(M^\sigma)^{-1}R) = b\rho((M^\sigma)^{-1}R^2) = \rho((M^\sigma)^{-1}(bM^\tau)) \leq 1, \\ \frac{1}{B}\rho(R^{-1}M^\sigma R^{-1}) = \frac{1}{B}\rho(R^{-2}M^\sigma) = \rho((BM^\tau)^{-1}M^\sigma) \leq 1. \end{aligned}$$

But since $\rho(\cdot) = \|\cdot\|_2$ for symmetric matrices, this implies

$$\|R(M^\sigma)^{-1}R\|_2 \|R^{-1}M^\sigma R^{-1}\|_2 \leq B/b,$$

so that

$$\frac{\|(M^\sigma)^{-1}\|_2}{\|R^{-1}\|_2^2} \cdot \frac{\|M^\sigma\|_2}{\|R\|_2^2} \leq \frac{B}{b}.$$

Finally, by the symmetry of R we have

$$\|R\|_2^2 = \rho^2(R) = \rho(R^2) = \rho(M^\tau) = \|M^\tau\|_2,$$

and similarly $\|R^{-1}\|_2^2 = \|(M^\tau)^{-1}\|_2$, so we must have

$$\|M^\sigma\|_2 \|(M^\sigma)^{-1}\|_2 \leq \frac{B}{b} \|M^\tau\|_2 \|(M^\tau)^{-1}\|_2,$$

as required. □

The Laplacian of a graph G (cf. [68]), denoted by $L(G)$, is the matrix M^τ when $\tau \equiv 1$. Theorem E.3 can yield useful bounds for J_{tp} when $\kappa_2(L(G))$ is known. For a Cartesian grid, it is well known that $\kappa_2(L(G)) = O(1/h^2)$; as a result, $\kappa_2(J_{tp})$ is also $O(1/h^2)$, provided the absolute permeability $K(x)$ and total mobility $\lambda_T(S)$ satisfy

$$\begin{gathered} 0 < k_{\min} \leq K(x) \leq k_{\max}, \\ 0 < \lambda_{T,\min} \leq \lambda_T(S) \leq \lambda_{T,\max}. \end{gathered}$$

Bibliography

[1] J. E. Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Model. Simul.*, 2:421–439, 2004.

[2] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Computat. Geosci.*, 6:405–432, 2002.

[3] J. R. Appleyard and I. M. Cheshire. Nested factorization. SPE paper 12264, presented at the SPE Symposium on Reservoir Simulation in San Francisco, CA, 1983.

[4] J. R. Appleyard and I. M. Cheshire. The cascade method for accelerated convergence in implicit simulators. In *European Petroleum Conference*, pp. 113–122, 1982.

[5] J. R. Appleyard, I. M. Cheshire, and R. K. Pollard. Special techniques for fully implicit simulators. In *Proc. European Symposium on Enhanced Oil Recovery*, pp. 395–408, Bournemouth, England, 1981.

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

[7] Z.-Z. Bai, G. H. Golub, and M. K. Ng. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, 24(3):603–626, 2002.

[8] A. Behie. Comparison of nested factorization, constrained pressure residual, and incomplete factorization preconditionings. SPE paper 13531, presented at the SPE Reservoir Simulation Symposium in Dallas, TX, 1985.

[9] J. B. Bell, C. N. Dawson, and G. R. Shubin. An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. *J. Comput. Phys.*, 74:1–24, 1988.

[10] M. Benzi, D. B. Szyld, and A. van Duin. Orderings for incomplete factorization preconditioning of nonsymmetric problems. *SIAM J. Sci. Comput.*, 20:1652–1670, 1999.

[11] M. Blunt and B. Rubin. Implicit flux limiting schemes for petroleum reservoir simulation. *J. Comput. Phys.*, 102(1):194–210, 1992.

[12] C. Bolley and M. Crouzeix. Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques. *RAIRO Anal. Numer.*, 12(3):237–245, 1978.

[13] Y. Brenier and J. Jaffré. Upstream differencing for multiphase flow in reservoir simulation. *SIAM J. Numer. Anal.*, 28(3):685–696, 1991.

[14] R. P. Brent. *Algorithms for Minimization Without Derivatives*, chapter 3–4. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[15] R. Bridson and C. Greif. A multipreconditioned conjugate gradient algorithm. *SIAM J. Matrix Anal. Appl.*, 27:1056–1068, 2006.

[16] H. Cao. *Development of Techniques for General Purpose Simulators.* PhD thesis, Stanford University, Stanford, CA, June 2002.

[17] H. Cao, H. A. Tchelepi, J. Wallis, and H. Yardumian. Parallel scalable unstructured CPR-type linear solver for reservoir simulation. SPE Paper 96809, presented at the SPE Annual Technical Conference and Exhibition in Dallas, TX, 2005.

[18] W. H. Chen, L. J. Durlofsky, B. Engquist, and S. Osher. Minimization of grid orientation effects through the use of higher order finite difference methods. *SPE Advanced Technology Series*, 1(2):43–52, 1991.

[19] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, 4(4):308–317, 2001.

[20] K. H. Coats. A note on IMPES and some IMPES-based simulation models. *SPE J.*, 5(3):245–251, Sept. 2000.

[21] K. H. Coats, W. D. George, and B. E. Marcum. Three-dimensional simulation of steamflooding. *Trans. SPE of AIME*, 257:573–592, 1974.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. D. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.

[23] M. G. Crandall and T. M. Liggett. Generation of semi-groups of nonlinear transformations on general Banach spaces. *Amer. J. Math.*, 93:265–298, 1971.

[24] M. G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Math Comp.*, 34:1–21, 1980.

[25] E. F. D'Azevedo, P. A. Forsyth, and W.-P. Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM J. Matrix Anal. Appl.*, 13(3):944–961, 1992.

[26] R. de Loubens. Construction of high-order adaptive implicit methods for reservoir simulation. Master's thesis, Stanford University, June 2007.

[27] K. Deimling. *Nonlinear Functional Analysis*. Springer-Verlag, 1985.

[28] J. E. Dennis, Jr., J. M. Martínez, and X. Zhang. Triangular decomposition methods for solving reducible nonlinear systems of equations. *SIAM J. Optimization*, 4:358–382, 1994.

[29] P. Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms.* Springer-Verlag, Berlin, 2004.

[30] I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29:635–657, 1989.

[31] L. J. Durlofsky and M. C. H. Chien. Development of a mixed finite-element-based compositional reservoir simulator. SPE Paper 25253, presented at the 12th SPE Symposium on Reservoir Simulation in New Orleans, LA, 1993.

[32] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, 1998.

[33] S. Evje and K. H. Karlsen. Degenerate convection-diffusion equations and implicit monotone difference schemes. In M. Fey and R. Jeltsch, editors, *Hyperbolic problems: Theory, Numerics, Applications*, volume 129, pp. 285–294. Birkhäuser Verlag, 1999.

[34] R. E. Ewing. Simulation of multiphase flows in porous media. *Transport Porous Med.*, 6:479–499, 1991.

[35] F. J. Fayers and J. D. Matthews. Evaluation of normalized Stone's methods for estimating three-phase relative permeabilities. *SPE J.*, 24:224–232, 1984.

[36] P. A. Forsyth and P. H. Sammon. Practical considerations for adaptive implicit methods in reservoir simulation. *J. Comput. Phys.*, 62:265–281, 1986.

[37] Geoquest. *Eclipse Technical Description 2005A.* Schlumberger, 2005.

[38] A. George and J. W. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1981.

[39] G. H. Golub and C. F. V. Loan. *Matrix Computations.* Johns Hopkins University Press, 3rd edition, 1996.

[40] A. Greenbaum, V. Pták, and Z. Strakoš. Any nonincreasing convergence curve is possible for GMRES. *SIAM J. Matrix Anal. Appl.*, 17:465–469, 1996.

[41] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 135:260–278, 1997.

[42] M. Honarpour, L. F. Koederitz, and H. A. Harvey. Empirical equations for estimating two-phase relative permeability in consolidated rock. *J. Petrol. Technol.*, 34:2905–2908, 1982.

[43] T. Y. Hou and X. H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[44] Y. Jiang. Tracer flow modeling and efficient solvers for GPRS. Master's thesis, Stanford University, June 2004.

[45] S. N. Kružkov. First order quasilinear equations in several independent variables. *Math. USSR Sbornik*, 10(2):217–243, 1970.

[46] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math*, 13:217–237, 1960.

[47] S. H. Lee, L. J. Durlofsky, M. F. Lough, and W. H. Chen. Finite difference simulation of geologically complex reservoirs with tensor permeabilities. *SPE Reserv. Eval. Eng.*, 1(6):567–574, 1998.

[48] S. H. Lee, H. A. Tchelepi, P. Jenny, and L. J. DeChant. Implementation of a flux-continuous finite-difference method for stratigraphic, hexahedron grids. *SPE J.*, 7(3):267–277, 2002.

[49] R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser, 2nd edition, 1992.

[50] B. J. Lucier. On nonlocal monotone difference schemes for scalar conservation laws. *Math. Comp.*, 47(175):19–36, 1986.

[51] R. C. MacDonald and K. H. Coats. Methods for numerical simulation of water and gas coning. *Trans. SPE of AIME*, 249:425–436, 1970.

[52] B. T. Mallison. *Streamline-based Simulation of Two-phase, Multicomponent Flow in Porous Media*. PhD thesis, Stanford University, 2004.

[53] S. F. Matringe, R. Juanes, and H. A. Tchelepi. Mixed-finite-element and related-control-volume discretizations for reservoir simulation on three-dimensional unstructured grids. SPE Paper 106117, presented at the SPE Reservoir Simulation Symposium in Houston, TX, 2007.

[54] A. Meister and C. Vömel. Efficient preconditioning of linear systems arising from the discretization of hyperbolic conservation laws. *Adv. Comp. Math.*, 14:49–73, 2001.

[55] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen. How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. Appl.*, 13:778–795, 1992.

[56] J. R. Natvig, K.-A. Lie, and B. Eikemo. Fast solvers for flow in porous media based on discontinuous Galerkin methods and optimal reordering. In *Computational Methods in Water Resources XVI*, 2006.

[57] A. S. Odeh. Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. *J. Petrol Technol.*, 33(1):13–25, 1981.

[58] F. M. Orr, Jr. *Theory of Gas Injection Processes*. Tie-Line Publications, 2007.

[59] J. M. Ortega and W. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.

[60] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235, 1984.

[61] D. W. Peaceman. A nonlinear stability analysis for difference equations using semi-implicit mobility. *SPE J.*, 17:79–91, 1977.

[62] H. S. Price and K. H. Coats. Direct methods in reservoir simulation. *Trans. SPE of AIME*, 257:295–308, 1974.

[63] S. C. Reddy and L. N. Trefethen. Pseudospectra of the convection-diffusion operator. *SIAM J. Appl. Math.*, 54:1634–1649, 1994.

[64] W. C. Rheinboldt. On M-functions and their application to nonlinear Gauss-Seidel iterations and to network flows. *J. Math. Anal. Appl.*, 32:274–307, 1970.

[65] H. L. Royden. *Real Analysis*. Prentice-Hall, 1988.

[66] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.

[67] T. F. Russell. Stability analysis and switching criteria for adaptive implicit methods based on the CFL condition. SPE paper 18416, presented at the SPE Symposium on Reservoir Simulation in Houston, TX, 1989.

[68] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.

[69] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7(3):856–869, July 1986.

[70] R. Sanders. On convergence of monotone finite difference schemes with variable spatial differencing. *Math. Comp.*, 40:91–106, 1983.

[71] A. Settari and K. Aziz. Treatment of nonlinear terms in the numerical solution of partial differential solutions for multiphase flow in porous media. *Int. J. Multiphase Flow*, 1:817–844, 1975.

[72] A. G. Spillette, J. G. Hillestad, and H. L. Stone. A high-stability sequential solution approach to reservoir simulation. SPE Paper 4542, presented at the Fall Meeting of the Society of Petroleum Engineers of AIME in Las Vegas, NV, 1973.

[73] H. L. Stone. Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM J. Numer. Anal.*, 5:530–568, 1968.

[74] K. Stueben. Algebraic multigrid (AMG): experiences and comparisons. In *Proceedings of the International Multigrid Conference*, Copper Mountain, CO, 1983.

[75] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time dependent problems. *Acta Numerica*, 12:451–512, 2003.

[76] G. W. Thomas and D. H. Trunau. Reservoir simulation using an adaptive implicit method. *SPE J.*, 23:759–768, 1983.

[77] B. van Leer. Upwind and high-resolution methods for compressible flow: from donor cell to residual-distribution schemes. *Commun. Comput. Phys.*, 1:192–206, 2006.

[78] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, 1962.

[79] S. Verma and K. Aziz. Control volume scheme for flexible grids in reservoir simulation. SPE paper 37999, presented at the SPE Symposium on Reservoir Simulation in Dallas, TX, 1997.

[80] P. K. W. Vinsome. ORTHOMIN, an iterative method for solving sparse banded sets of simultaneous linear equations. SPE paper 5729, presented at the SPE Symposium on Numerical Simulation of Reservoir Performance in Los Angeles, CA, 1976.

[81] J. R. Wallis, R. P. Kendall, and T. E. Little. Constrained residual acceleration of conjugate residual methods. SPE paper 13536, presented at the SPE Reservoir Simulation Symposium in Dallas, TX, 1985.

[82] J. W. Watts. A compositional formulation of the pressure and saturation equations. *SPE Reservoir Eng.*, 1(3):243–252, 1986.

[83] J. W. Watts. Reservoir simulation: past, present and future. *SPE Computer Applications*, 12(4):171–176, 1997.

[84] J. W. Watts III. A conjugate gradient truncated direct method for the iterative solution of the reservoir simulation pressure equation. *SPE J.*, 21:345–353, 1981.

[85] D. M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[86] L. C. Young. A finite-element method for reservoir simulation. *SPE J.*, 21(1):115–128, 1981.

[87] L. C. Young and R. E. Stephenson. A generalized compositional approach for reservoir simulation. *SPE J.*, 23(5):727–742, 1983.