

ORDERING-BASED NONLINEAR SOLVER WITH ADAPTIVE
COUPLING FOR MULTIPHASE FLOW AND TRANSPORT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF ENERGY
RESOURCES ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Mohammad Shahvali

November 2012

© Copyright by Mohammad Shahvali 2013
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.

(Ronaldo Borja)

Approved for the University Committee on Graduate Studies.

Abstract

The Fully Implicit Method (FIM) is widely used to discretize the nonlinear conservation equations that govern multiphase flow in porous media. Although FIM is unconditionally stable, the nonlinear solver, which is typically based on Newton's method, often fails to converge for large time steps. As the time step size increases, the complexity of the nonlinear coupling increases. In the presence of strong buoyancy in a heterogeneous domain, for example, counter-current flow may be present, and phase-flow reversal - as a function of time or Newton iteration - becomes an issue. It does not take too many interfaces with flow reversal across them to make the nonlinear solver stall. Even when the Newton-based iterative procedure converges, the cost associated with updating all the unknowns simultaneously - everywhere and all the time - can be quite expensive. Conventional sequential-implicit strategies can be used to reduce the cost, but they suffer from severe restrictions on the allowable timestep size. We propose a new nonlinear solution strategy to improve the robustness and efficiency of FIM. Specifically, for each FIM Newton iteration, we update pressure by solving a linear system obtained from the full Jacobian. Then, we construct a directed graph with grid-cells as nodes and phase fluxes at grid-cell interfaces as edges, to identify the counter-current flow regions. After that, we employ a phase-based potential ordering that makes it possible to visit the components once, from upstream to downstream. The graph components vary in size from a single cell to multiple 'strongly

connected' cells. For each component made up of a single cell, we update the saturation nonlinearly. For each component made up of multiple cells (strongly connected region), we perform a simultaneous linear update of both the saturation and pressure. The saturation iterates are safeguarded based on 'trust regions' of the flux (fractional flow) functions. We compare the new solver with the current state-of-the-art for a wide range of heterogeneous problems with special focus on counter-current flow due to buoyancy. Our results show that the proposed nonlinear FIM solver converges for arbitrarily large time steps and with excellent convergence rates.

Contents

Abstract	iv
1 Introduction	1
1.1 Standard black-oil model	2
1.1.1 Incompressible two-phase flow	4
1.2 Discretization of the governing equations	5
1.2.1 Discretization in space	6
1.2.2 Discretization in time	8
1.3 Nonlinear solvers	11
1.3.1 Standard Newton's method	11
1.3.2 Newton's method with heuristic safeguards	12
1.3.3 Continuation Newton	14
1.3.4 Ordering-based methods	15
1.4 Thesis overview	17
2 Newton Safeguard Based on Flux Function	19
2.1 Nonlinearity of transport equation	20
2.2 Convergence maps	21
2.2.1 Concave flux	22
2.2.2 Convex flux	23

2.2.3	S-shaped flux	24
2.2.4	Flux in the presence of gravity	26
2.3	Newton safeguard based on trust regions of flux function	27
2.3.1	Safeguard for S-shaped flux function	27
2.3.2	Extension of the safeguard algorithm for flux functions with gravity	28
2.4	The fully coupled FIM with Newton safeguard	30
2.5	Numerical examples	33
2.5.1	Example 1: Top-down gravity segregation with injection from the top (1-D)	33
2.5.2	Example 2: Homogeneous quarter 5-spot with gravity	36
2.5.3	Example 3: Top-down gravity segregation with injection in a homogeneous quarter 5-spot	38
2.5.4	Example 4: Lock-exchange with injection in a homogeneous quarter 5-spot	40
2.5.5	Example 5: Top-down gravity segregation with injection in a heterogeneous quarter 5-spot	42
2.5.6	Example 6: Lock-exchange with injection in a heterogeneous quarter 5-spot	45
3	Nonlinear Coupling of Flow and Transport	47
3.1	Convergence of pressure and saturation in the fully coupled FIM . . .	48
3.2	Effect of selective pressure updating on convergence of the fully coupled FIM	49
3.2.1	Updating pressure only in the first few Newton iterations . . .	50
3.2.2	Updating pressure only where total velocity changes exceed a specified threshold	51

3.3	Phase-based ordering	59
3.4	Reduced Newton method	61
3.4.1	Convergence behavior of reduced Newton method	62
4	Efficient Coupling of Flow and Transport	66
4.1	Graph theory	67
4.2	Phase flow graph	69
4.2.1	Pure co-current flow	70
4.2.2	Pure counter-current flow	71
4.2.3	Mixed flow	72
4.3	Adaptively coupled hybrid method (ACHM)	73
4.3.1	Pressure update	74
4.3.2	Identifying strongly connected regions	74
4.3.3	Sequential saturation update and safeguard	75
4.4	Numerical examples	78
4.4.1	Homogeneous 2D reservoir with gravity - uniform initial saturation condition	78
4.4.2	Heterogeneous 2D reservoir with gravity - uniform initial saturation condition	83
4.4.3	Top-down gravity segregation with slight injection in a heterogeneous 2D reservoir	87
4.4.4	Waterflood in a large heterogeneous 3D reservoir	90
5	Conclusions and Future Directions	93
5.1	Conclusions	93
5.2	Future directions	95
5.2.1	Extending ACHM to three-phase models	96
5.2.2	Extending ACHM to compositional models	96

List of Tables

2.1	Rock and fluid properties used in example 1.	34
4.1	Number of single-cell components and size of the largest multi-cell component at different iterations (normallized by the total number of grid cells, i.e. 2500).	82
4.2	Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 2500).	86
4.3	Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 2500).	89
4.4	Comparison between the number of iterations required for convergence of the fully coupled FIM with safeguard and ACHM. For most timesteps, ACHM needs fewer or the same number of iterations. . . .	91
4.5	Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 138600), corresponding to the final timestep, $\Delta t = 50$ days.	92

List of Figures

1.1	Effect of Appleyard chop on convergence of Newton's method for solving a 1D Buckley Leverett displacement problem using the fully implicit method (FIM). standard Newton's method does not converge for timesteps larger than 10^{-2} PVI as shown by the dashed line. The modified Appleyard chop results in unconditional convergence of Newton's method by limiting saturation changes to 0.2 over iterations.	14
2.1	Effect of Courant number on shape of residual function corresponding to single-cell water injection problem. The intersection of the dashed line with each residual function represents the solution. The figure shows that for large timesteps (large Courant numbers), nonlinear behavior of residual function is influenced by shape of fractional flow function.	22
2.2	A typical concave flux function shown on the left, obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 0.1$. The corresponding convergence map is shown on the right. Newton's method is unconditionally convergent in the case of concave flux functions.	23

2.3	A typical convex flux function shown on the left, obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 10$. The corresponding convergence map is shown on the right. Newton's method is unconditionally convergent in the case of convex flux functions.	24
2.4	A typical S-shaped flux function shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$ and $\frac{\mu_w}{\mu_o} = 0.1$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas; however convergence is always guaranteed if inflection point is chosen as the initial guess.	25
2.5	A typical S-shaped flux function shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$ and $\frac{\mu_w}{\mu_o} = 10$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas; however convergence is always guaranteed if inflection point is chosen as the initial guess.	25
2.6	A typical flux function in the presence of gravity shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$, $\frac{\mu_w}{\mu_o} = 0.1$ and $N_g = -2$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas of the convergence map.	26
2.7	Convergence maps for S-shaped flux functions (presented in Figures 2.4 and 2.5), obtained using the Newton safeguard. Convergence is obtained everywhere in the $(f_w^* - S_w^{(0)})$ plane.	28
2.8	Convergence map for flux function with gravity (presented in Figure 2.6), obtained using the proposed Newton safeguard. Convergence is obtained everywhere in the $(f_w^* - S_w^{(0)})$ plane.	30

2.9	Saturation profile at $t = 10$ cell PVI, corresponding to example 1, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 10$ cell PVI.	34
2.10	The CFL number versus dimensionless height, at $t = 10$ cell PVI, corresponding to example 1, obtained using the fully coupled FIM with the flux function-based Newton safeguard.	35
2.11	Number of iterations versus timestep size corresponding to example 1. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 1$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	35
2.12	Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 2, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.	37
2.13	The CFL map, at $t = 1250$ cell PVI corresponding to example 2, obtained using the fully coupled FIM with the flux function-based Newton safeguard.	37
2.14	Number of iterations versus timestep size corresponding to example 2. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 1$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	38

2.15	Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 3, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.	39
2.16	The CFL map, at $t = 1250$ cell PVI corresponding to example 3, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.	39
2.17	Number of iterations versus timestep size corresponding to example 3. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 30$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	40
2.18	Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 4, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.	41
2.19	The CFL map, at $t = 1250$ cell PVI corresponding to example 4, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.	41
2.20	Number of iterations versus timestep size corresponding to example 4. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 3$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	42
2.21	Log-normal permeability field used for heterogeneous examples. . . .	43

2.22	Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 5, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.	43
2.23	The CFL map, at $t = 1250$ cell PVI corresponding to example 5, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.	44
2.24	Number of iterations versus timestep size corresponding to example 5. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t > 15$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	44
2.25	Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 6, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.	45
2.26	The CFL map, at $t = 1250$ cell PVI corresponding to example 4, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.	46
2.27	Number of iterations versus timestep size corresponding to example 6. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t > 5$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.	46

3.1	Maximum change of pressure iterates (left) and saturation iterates (right) over Newton iteration for the small test problem. The pressure iterates (corresponding to elliptic flow equation) tend to converge to solution faster than saturation iterates (corresponding to hyperbolic transport equation).	49
3.2	Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.	51
3.3	The figures on the left and right show the phase flow directions at the beginning and at the end of a hypothetical FIM iteration respectively. The phase flow reversal over Newton iterations is sometimes referred to as phase flip flop. This phenomenon results in discontinuity of flux function over iterations due to change in the upstream directions and could lead to oscillations in residual and convergence failure.	53
3.4	Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.	56
3.5	The map of cells that have phase flip flop at the corresponding interfaces for six sample Newton iterations, $\nu = 2, 4, 18, 20, 23$ and 26. In early Newton iterations, phase flip flops occur more frequently but phase flow directions tend to become steady as we approach solution in later iterations.	57

3.6	The map of cells that have variations of total velocities at the corresponding interfaces greater than the specified threshold, $\epsilon = 0.001$ for six sample Newton iterations, $\nu = 2, 4, 18, 20, 23$ and 26 . These are the cells that require pressure updates. As we proceed with the nonlinear iterations, fewer pressure updates are needed.	57
3.7	L_∞ norm of the residual of saturation equations with respect to the iteration number. Initially the residual has oscillations due to phase flip flops. As soon as flow directions become steady (marked by the arrow), the residual begins to decline and eventually it becomes stagnant. The final residual norm in this example with $\epsilon = 0.001$, happens to meet the convergence criterion and therefore convergence is obtained. . . .	58
3.8	Plot of parameter ϵ versus the stabilized L_∞ norm of the residual, referred to as final error. By specifying a greater ϵ , the number of pressure updates in each iteration decreases, since a greater change in total velocity is required to update pressure. Less frequent pressure updates give rise to stagnant residuals with larger L_∞ norms.	58
3.9	Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.	64
3.10	Saturation map at $t = \Delta t = 1000$ cell PVI, corresponding to the displacement problem with no gravity effects (pure co-current flow), obtained using reduced Newton method. For this extremely large timestep, convergence is obtained in 4 iterations.	65

3.11	Saturation map at $t = \Delta t = 10$ cell PVI, corresponding to the displacement problem with gravity effects due to density difference of fluids, obtained using reduced Newton method. The timestep used here is the largest for which convergence is obtained.	65
4.1	The figure shows a directed graph on the left and an undirected graph on the right. Both graphs have the same number of nodes and edges, but have entirely different structures.	67
4.2	The adjacency list (on the left) and the adjacency matrix (on the right) representations of the directed graph presented in Figure 4.1.	68
4.3	In the case of pure co-current flow, each cell is a strongly connected component of phase flow graph (shown in grey).	71
4.4	In the case of pure counter-current flow, only one strongly connected component exists (shown in grey), which is made up of the entire grid cells.	71
4.5	In the case of mixed flow where co-current and counter-current flow interfaces coexist, strongly connected components vary from single-cell to multi-cell components. The phase flow graph shown in this example is decomposed into five strongly connected components, two of which are made up of multiple cells and the rest are single-cell components indicating counter-current and co-current flow regions respectively. . .	72
4.6	At each Newton iteration, the adaptively coupled hybrid method (ACHM) consists of three consecutive steps.	74
4.7	The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.	80

4.8	The figure shows maps of strongly connected components corresponding to different iterations. In each map, the dark blue area represents the entire single-cell components, whereas each multi-cell component is portrayed using a different color. At each iteration, a significant part of the domain is composed of single-cell components and multi-cell components are generally small in size.	81
4.9	Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.	82
4.10	The log-normal heterogeneous permeability field used for testing ACHM.	84
4.11	The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.	84
4.12	The figure shows the map of strongly connected components corresponding to iteration $\nu = 9$ of $\Delta t = 100$ cell PVI. In this map, the dark blue area represents the entire single-cell components, whereas each multi-cell component is portrayed using a different color. It can be inferred that significant part of the domain is composed of single-cell components and multi-cell components are small in size.	85
4.13	Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.	86
4.14	The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.	88

4.15 Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.	88
4.16 Permeability field and well configuration for the $2 \times 2 \times 2$ upscaled SPE 10 model.	92

Chapter 1

Introduction

Petroleum reservoir simulation is the use of numerical methods to solve the equations that govern heat and fluid flow in porous media. Today, reservoir simulation plays a key role in the oil and gas industry. Major decisions are made fully or at least partially based on some sort of reservoir simulation study. Growth of computing power and data storage capacity has enabled us to simulate very large models in the order of up to one billion grid cells (Dogru et al. (2008)). Not only has the size of reservoir simulation models increased dramatically, but simulation studies aim to address more complex physical phenomena. The ever-increasing complexity of simulation studies is due to highly nonlinear and tightly coupled governing equations as well as the inherent geological complexity of earth models. As a result of increase in complexity and size of models, efficient simulation of displacement processes in porous media remains a major challenge and an ongoing research topic.

The governing equations for multiphase flow in porous media are based on conservation of mass for different components. Reservoir simulation models are categorized into two groups, standard black-oil models and compositional models (Aziz and Settari (1979)). In compositional models, conservation equations are written for individual components (Young and Stephenson (1983)). Despite the increase in the use

of compositional models, high computational cost associated with phase equilibrium calculations in these models still remains a major drawback. On the other hand, the simplifying assumptions regarding chemical components and thermodynamic models make standard black-oil models attractive candidates for most reservoir simulation studies in the industry.

The remainder of this chapter is organized as follows. We first introduce the governing equations for the standard black-oil model. We then review the discretization schemes, with special focus on finite volume method for discretization of the governing equations in space and time. Next, we provide a brief literature review of some of the most commonly used nonlinear solvers in reservoir simulation. Finally, we present an overview of the thesis.

1.1 Standard black-oil model

In a standard black-oil model, certain assumptions are made to simplify the general mass balance equations. First, the chemical species are represented by three pseudo-components, water, oil and gas which exist in aqueous, liquid and vapor hydrocarbon phases. Secondly, it is assumed that the water and the oil components exist only in the aqueous phase and liquid hydrocarbon phase respectively, whereas the gas component phase can be present in the vapor and the liquid hydrocarbon phases. Finally, solubility of gas in the liquid hydrocarbon phase is assumed to be a function of gas pressure. With these assumptions, the mass balance equations for different components take the form (Aziz and Settari (1979))

$$\frac{\partial (\phi \rho_p S_p)}{\partial t} + \nabla \cdot (\rho_p \mathbf{u}_p) = \rho_p q_p, \quad (1.1)$$

for $p = o, w$, and

$$\frac{\partial (\phi \rho_g S_g + \phi \rho_o S_o R_s)}{\partial t} + \nabla \cdot (\rho_g \mathbf{u}_g + \rho_o \mathbf{u}_o R_s) = \rho_g q_g, \quad (1.2)$$

for the gas component, where ϕ , ρ , S , R_s , q and $\mathbf{u}$ denote porosity, density, phase saturation, solubility ratio, source term and phase velocity respectively. The phase velocities $\mathbf{u}_p$, are obtained from generalized Darcy's law as follows

$$\mathbf{u}_p = \mathbf{K} \lambda_p (\nabla (p_p - \gamma_p z)), \quad (1.3)$$

where $\mathbf{K}$, p_p , γ_p and z represent permeability tensor, phase pressure, specific weight and depth respectively. In an isotropic permeability case, the permeability tensor can be written as $\mathbf{K} = K \mathbf{I}$. The phase mobility, λ_p , is defined as

$$\lambda_p = \frac{k_{rp}}{\mu_p}. \quad (1.4)$$

where k_{rp} and μ_p are phase relative permeability and viscosity respectively. The relative permeabilities are generally defined as functions of phase saturations (Fayers and Matthews (1984)),

$$k_{rw} = k_{rw}(S_w), \quad k_{rg} = k_{rg}(S_g), \quad k_{ro} = k_{ro}(S_o, S_g). \quad (1.5)$$

The capillary pressure constraints relate the phase pressures as follows

$$p_o - p_w = P_{cow}(S_w), \quad p_g - p_o = P_{cgo}(S_g). \quad (1.6)$$

Both relative permeability and capillary pressure have highly nonlinear dependence on saturation variables, giving rise to nonlinear behavior of the governing equations.

1.1.1 Incompressible two-phase flow

Throughout this thesis, we often address the incompressible flow problem for a water-oil system with no capillarity. The corresponding governing transport equations in this case can be readily derived from the general black-oil model by considering constant phase densities and zero capillary pressure in Equation 1.1. The final equation is

$$\phi \frac{\partial (S_p)}{\partial t} + \nabla \cdot (\lambda_p K \nabla (p - \gamma_p z)) = q_p, \quad p = o, w. \quad (1.7)$$

It is sometimes useful to reformulate transport equations in terms of fractional flow (flux) and total velocity functions. Total velocity is obtained by summing up individual phase velocities, and has the form

$$\mathbf{u}_T = \lambda_T \nabla p - \sum_{p=o,w} \lambda_p \rho_p \mathbf{g}, \quad (1.8)$$

where λ_T is the total mobility obtained by summing up individual phase mobilities.

Fractional flow for each phase is the ratio of the phase volumetric flux to the total flux. Using Darcy's law to express phase velocities, the fractional flow takes the form

$$f_w = \frac{\lambda_w}{\lambda_T} (1 - N_g k_{ro}), \quad (1.9)$$

where N_g is the gravity number, denoting the ratio of gravitational forces to viscous forces, and is defined as

$$N_g = \frac{K (\rho_w - \rho_o) g}{\mu_o u_T}. \quad (1.10)$$

Using the above relations, transport equation for water phase can be written in the form

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w \mathbf{u}_T) = q_w. \quad (1.11)$$

Equation 1.11 is a nonlinear hyperbolic PDE in S_w . The nonlinearity of transport

equation is due to the nonlinear behavior of the fractional flow function.

Another way of expressing mass conservation equations is to reformulate them in the form of flow (pressure) and transport (saturation) equations. The pressure equation is a total mass balance, and is essentially a linear combination of the individual transport equations. In the case of incompressible two-phase flow, the pressure equation is obtained by summing up oil and water transport equations, and considering the saturation constraint, $S_w + S_o = 1$. The pressure equation in this case takes the form

$$\nabla \cdot (\mathbf{u}_T) = q_T, \quad (1.12)$$

or equivalently

$$\nabla \cdot (\lambda_T \nabla p - \sum_{p=o,w} \lambda_p \rho_p \mathbf{g}) = q_T, \quad (1.13)$$

where q_T is the total (oil and water) source term. Equation 1.13 is an elliptic PDE in p .

1.2 Discretization of the governing equations

The governing equations in the field of fluid flow in porous media are generally nonlinear and coupled through various terms. The complexity of these equations leads to lack of analytical solutions for most problems of interest. In order to solve these equations numerically, they need to be discretized in space and time. Finite element and finite volume-based discretization methods have been used in reservoir simulation. Although finite element methods offer tremendous flexibility in terms of treatment of unstructured grids, irregular boundaries and tensorial permeability fields, they are not as widely used as finite volume methods (Young (1981), Hou and Wu (1997) and Aarnes (2004)). Local mass conservation makes finite volume methods attractive candidates for reservoir simulation. The majority of general-purpose reservoir

simulators in industry are based on finite volume methods.

1.2.1 Discretization in space

In order to describe finite volume method for spatial discretization of flux terms in flow or transport equations, we focus on the following general PDE,

$$\nabla \cdot \mathbf{u} = f, \quad \text{in } \Omega, \quad (1.14)$$

where $\mathbf{u}$ could denote phase velocity or total velocity and f is a source term. Let $\Omega \subset \mathbb{R}^n$ be the reservoir domain in dimension n . The simulation grid represents a partition of Ω into a set of N non-overlapping control volumes Ω_i , $i = 1, \dots, N$. Each partition Ω_i is also called a grid cell. Integrating Equation 1.14 over each control volume Ω_i gives

$$\int_{\Omega_i} \nabla \cdot \mathbf{u} \, d\Omega = \int_{\Omega_i} f \, d\Omega. \quad (1.15)$$

Using the divergence theorem, the flux terms in Equation 1.15 can be expressed in the form of an integral over the boundary of the control volume, $\partial\Omega_i$, as follows

$$\int_{\partial\Omega_i} \mathbf{u} \cdot \mathbf{n}_i \, d\Gamma = \int_{\Omega_i} f \, d\Omega, \quad (1.16)$$

where $\mathbf{n}_i$ is the outward-pointing unit normal to $\partial\Omega_i$. For simplicity, we replace for velocity from Darcy's law without the gravity terms, we write

$$\int_{\partial\Omega_i} \lambda (\mathbf{K} \nabla p) \cdot \mathbf{n}_i \, d\Gamma = \int_{\Omega_i} f \, d\Omega. \quad (1.17)$$

Let J_i denote the set of indices j , such that control volumes Ω_j share an interface Γ_{ij} with control volume Ω_i . The total flux can be written as

$$\int_{\partial\Omega_i} \lambda (\mathbf{K} \nabla p) \cdot \mathbf{n}_i \, d\Gamma \approx \sum_{j \in J_i} \lambda_{ij} \int_{\Gamma_{ij}} (\mathbf{K} \nabla p) \cdot \mathbf{n}_{ij} \, d\Gamma_{ij}. \quad (1.18)$$

The approximation is due to the assumption that in the neighborhood of the interface Γ_{ij} , the mobility λ_{ij} is constant. Thus it can be taken out of the integral.

The challenge in the finite volume method is to find a good approximation for the flux terms on Γ_{ij} . A *two-point flux approximation (TPFA)* is used to approximate the flux at an interface shared between two control volumes, to the pressures of those control volumes. It is assumed that pressure varies linearly within each of the two control volumes. This assumption, in addition to continuity of pressure and flux at the interface, results in the following relation for the flux between two adjacent grid cells in 1D (e.g. in x direction),

$$F_{12} = \lambda (T_1 p_1 + T_2 p_2), \quad (1.19)$$

where the transmissibility coefficients T_1 and T_2 are expressed as

$$T_1 = -T_2 = \frac{2}{\Delta x_1/k_x^1 + \Delta x_2/k_x^2}. \quad (1.20)$$

For multidimensional problems, TPFA is based on a dimensional split.

In the general case of full tensor permeabilities on unstructured grids, a *multi-point flux approximation (MPFA)* can be used to approximate the total flux of a control volume by relating it to the pressure of the control volumes in the surrounding region.

Using an MPFA method,

$$\int_{\Gamma_{ij}} (\mathbf{K} \nabla p) \cdot \mathbf{n}_{ij} \, d\Gamma_{ij} \approx \sum_{k \in \Lambda_{ij}} T_k p_k, \quad (1.21)$$

where Λ_{ij} denotes the set of indices corresponding to control volumes that touch the interface Γ_{ij} (share at least a vertex with the interface), and T_k represents the transmissibility coefficient. For more information and details on MPFA methods, see Avatsmark (2002), Edwards and Rogers (1998) and Edwards (2002).

An important feature of the spatial discretization is approximating saturation-dependent terms at cell interfaces. In reservoir simulation, *phase-based upstream weighting* are commonly used for this purpose. Let i and $i + 1$ denote the indices of two adjacent cells in 1D, numbered from left to right. The mobility of phase p at the interface $i + 1/2$ shared by the cells is expressed as (Brenier and Jaffre (1991))

$$\lambda_{p,i+1/2} = \begin{cases} \lambda_p(S_i), & \text{if } u_{p,i+1/2} > 0, \\ \lambda_p(S_{i+1}), & \text{if } u_{p,i+1/2} < 0. \end{cases} \quad (1.22)$$

Notice that phase-based upwinding used in reservoir simulation, gives rise to numerical flux functions different from those obtained by other schemes used in CFD, such as the Godunov and Engquist-Osher schemes (LeVeque (1992)).

1.2.2 Discretization in time

Discretization in time plays a key role in the numerical stability of the discretization scheme. In the following, we briefly describe a number of temporal discretization schemes which are widely and commonly used in reservoir simulation.

Fully implicit method (FIM)

Based on the fully implicit method, all of the variables (e.g. pressure and saturation) at each cell are taken implicitly, i.e. at the new time level. The advantage of using FIM is the unconditional stability offered by the implicit temporal discretization (Aziz and Settari (1979), Peaceman (1977) and Settari and Aziz (1975)).

Adaptive implicit method (AIM)

Thomas and Trunau (1983) introduced an adaptive implicit strategy, based on which the cells are treated differently in terms of level of implicitness. For each cell, depending on the CFL number, implicit or explicit pressure and saturation variables are considered. Russell (1989) performed a stability analysis and proposed switching criteria (implicit to explicit and vice versa) for AIM based on CFL numbers. More details on practical considerations of AIM in reservoir simulation are presented in Forsyth and Sammon (1986).

Implicit pressure explicit saturation (IMPES)

Saturation-dependent and pressure-dependent terms in pressure equation are evaluated at the old and new time levels, n and $n + 1$, respectively, and the pressure equation is solved for p^{n+1} . Phase velocities are computed using S^n and p^{n+1} . Saturation is obtained by solving the mass balance equations while saturation and saturation-dependent terms are treated explicitly. Despite low computational cost of IMPES, there are severe restrictions on the allowable timestep sizes, since the method is only conditionally stable due to explicit treatment of saturation. More details on IMPES and the corresponding CFL conditions can be found in Coats (2000).

Sequential implicit method (SEQ)

Sequential-implicit separates the solution of the pressure equation, from that of the saturation equations. The pressure equation is solved first, by evaluating saturation-dependent terms explicitly and pressure-dependent terms implicitly (similar to IMPES). The total velocity field is computed by adding individual phase velocities evaluated using S^n and p^{n+1} . The total velocity field is frozen and the transport equations are solved while saturation-dependent terms are treated implicitly. The pressure update-saturation update steps are repeated until convergence in the outerloop is obtained.

McDonald and Coats (1970) showed that the use of implicit transmissibilities in the IMPES model for the simulation of well coning behavior, results in a several-fold increase in the allowable timestep size as compared to the IMPES scheme, while the computing time per step is increased by less than 10 percent.

Spillette et al. (1973) used a sequential implicit solution strategy for a variety of cases including counter-current imbibition, solution-gas drive and gas percolation and coning problems.

Coats et al. (1974) presented a sequential formulation for three-dimensional modeling of steam injection processes. They showed that the use of implicit transmissibilities results in stability of the method.

Since transport equations are solved implicitly, SEQ is unconditionally stable, provided that the total velocity field is conservative. However, for most problems with tight coupling between flow and transport, e.g. the ones with counter-current flow due to gravity, convergence of the outerloop is not guaranteed for large timesteps (Jenny et al. (2005) and Lunati and Jenny (2007)). As a result, SEQ suffers from severe restrictions on the allowable timestep size. Moreover, mass conservation error for one of the phases is present in compressible flow problems (Aziz and Settari (1979)).

1.3 Nonlinear solvers

A fully implicit finite volume discretization of the pressure and saturation equations for each grid cell, outlined in 1.13 and 1.11, leads to a nonlinear system of equations, referred to as the residual equations. Equation 1.23 shows a general form of the nonlinear residual system that needs to be solved at each timestep of an implicit simulation. In this equation, $\mathbf{x}^n \in \mathbb{R}^N$ and $\mathbf{x}^{n+1} \in \mathbb{R}^N$ denote the unknowns at the old and new states (timesteps) respectively. In a two-phase flow simulation, for instance, the unknowns include pressure and saturation variables of each grid cell.

$$\mathbf{R}(\mathbf{x}^{n+1}; \Delta t, \mathbf{x}^n) = 0, \quad (1.23)$$

In general, the residual equations are nonlinear due to various sources of nonlinearity including saturation-dependent nonlinear terms such as relative permeability and capillary pressure functions, or pressure-dependent nonlinear terms such as viscosities and densities. To solve the nonlinear system of equations, we need a strategy, often times referred to as a ‘nonlinear solver’. In the following, we describe the most commonly used nonlinear solvers in reservoir simulation.

1.3.1 Standard Newton’s method

Newton’s method is an iterative process, through which a sequence of iterates is generated by solving a linear system of equations at each iteration, obtained from linearization of the original nonlinear system 1.23. At each Newton iteration we solve

$$\mathbf{J}^\nu \delta \mathbf{x}^\nu = -\mathbf{R}^\nu \quad (1.24)$$

where $\mathbf{J}$ denotes the Jacobean matrix, representing the derivatives of the residual with respect to unknowns, and $\delta \mathbf{x}^\nu$ is the vector of Newton updates. Thus, the sequence

of iterates is generated starting from the old state, as follows

$$\begin{aligned}\mathbf{x}^{n+1,0} &= \mathbf{x}^n, \\ \mathbf{x}^{n+1,\nu+1} &= \mathbf{x}^{n+1,\nu} - \mathbf{J}^{-1}\mathbf{R}(\mathbf{x}^{n+1,\nu}; \Delta t, \mathbf{x}^n), \quad \nu = 0, 1, \dots\end{aligned}\tag{1.25}$$

Convergence limitation of Newton's method is best understood by considering a particular dynamical system for which Newton iteration is an explicit first-order timestep-ping scheme. The dynamical system is defined as (Deuffhard (2004) and Younis (2011))

$$\begin{aligned}\mathbf{x}^{n+1} &= \mathbf{x}^n, & \nu &= 0, \\ \frac{d\mathbf{x}^{n+1}}{d\nu} &= -\mathbf{J}^{-1}\mathbf{R}(\mathbf{x}^{n+1}; \Delta t, \mathbf{x}^n), & \nu &> 0.\end{aligned}\tag{1.26}$$

Notice that in the above equation, the Newton iteration index, ν , is considered to be a continuous quantity. A first-order explicit discretization of Equation 1.26 results in the following discrete form representing the dynamical system

$$\mathbf{x}^{n+1,\nu+1} - \mathbf{x}^{n+1,\nu} = -\Delta\nu\mathbf{J}^{-1}\mathbf{R}(\mathbf{x}^{n+1,\nu}; \Delta t, \mathbf{x}^n).\tag{1.27}$$

Comparison of Equations 1.27 and 1.25 reveals that Newton's method approximates the derivative of the new state with respect to the embedded time, ν , using a first-order finite-difference scheme with a unit step size, $\Delta\nu = 1$ (Deuffhard (2004) and Younis (2011)). Since explicit first-order timestepping may be unstable (has a timestep restriction), Newton iterations may not converge, even though the continuous Newton flow is well-behaved.

1.3.2 Newton's method with heuristic safeguards

Heuristic approaches are used in commercial simulators to improve convergence behavior of standard Newton's method. One of the most commonly used methods is the so-called *modified Appleyard chop algorithm* (e.g. used in *EclipseTM*) which is a

cell-based approach. For any grid cell, given the Newton saturation iterates at the old and new iteration levels, the algorithm reads as follows (Geoquest (2005) and Younis (2011)),

- If saturation has changed from immobile to mobile over the iteration, scale back the saturation change such that it is barely mobile.
- If saturation has changed from mobile to immobile over the iteration, scale back the saturation change such that it is barely mobile.
- Ensure that saturations are between 0 and 1.
- Ensure that saturation change over the iteration is limited to a specified small number (usually 0.2). If saturation change exceeds that amount, scale it back.

The following example illustrates how the Appleyard chop algorithm improves the convergence behavior of standard Newton's method for a Buckley-Leverett one-dimensional displacement problem. Consider a horizontal porous medium initially filled with oil. Water is injected from the left boundary at a constant rate to displace the resident oil, and fluids are produced from the right boundary at a constant pressure. The rock and fluids are incompressible. We use a fully implicit discretization to solve this displacement problem using 100 grid cells. We employ standard Newton's method to solve the nonlinear problem for only one timestep using a range of timestep sizes. Figure 1.1 shows the number of iterations required for convergence versus dimensionless timestep size (pore volumes injected, PVI). The results show that standard Newton's method does not converge for timesteps greater than 10^{-2} PVI. However, unconditional convergence regardless of timestep size, is obtained if saturation changes in the cells are limited to 0.2.

Another heuristic safeguard for Newton's method in black oil simulation is the *geometric penalty*, which is very similar to the Appleyard chop method except that

the saturation changes are limited to 20% of the original saturation (Naccache (1997)).

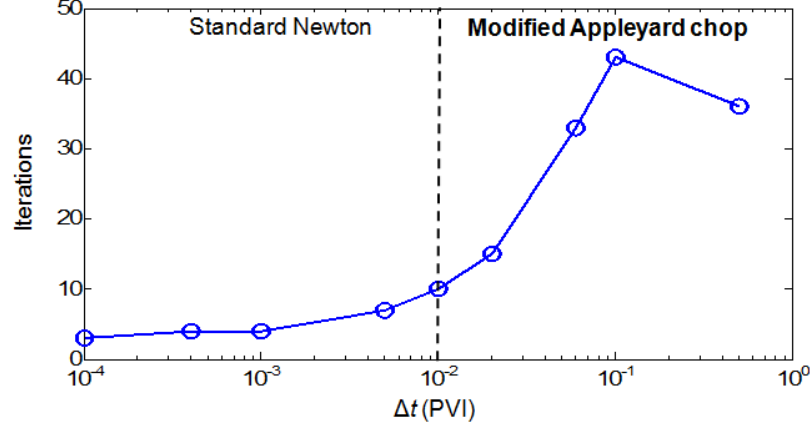


Figure 1.1: Effect of Appleyard chop on convergence of Newton’s method for solving a 1D Buckley Leverett displacement problem using the fully implicit method (FIM). standard Newton’s method does not converge for timesteps larger than 10^{-2} PVI as shown by the dashed line. The modified Appleyard chop results in unconditional convergence of Newton’s method by limiting saturation changes to 0.2 over iterations.

1.3.3 Continuation Newton

Younis and Tchelepi (2009), formulated a continuation-based solution process that associates a timestep size with each iteration. Each iterate is a pair of the unknown state augmented with a corresponding timestep. The iteration sequence follows along a solution path, parameterized using a single parameter, toward the solution of the target timestep size. The initial iterate consists of the initial state and timestep of zero. To increase the computational efficiency and to avoid following the solution path too closely, a convergence neighborhood around the solution path is considered. In case the iterates exit the convergence neighborhood, they are directed back into it using a single Newton step.

1.3.4 Ordering-based methods

Ordering-based methods, aim to reorder the grid cells or equations and variables, to allow decoupling of equations (Appleyard and Cheshire (1982), Kwok and Tchelepi (2007), Natvig et al. (2006), Natvig and Lie (2008), and J. E. Dennis and Zhang (1994)). There are two classes of ordering-based methods, including cell-based and phase-based approaches. In cell-based methods, the cells are ordered along the flow direction, and transport equations are solved on a cell-by-cell basis in a sequential fashion. In phase-based methods, ordering is performed based on phase potentials, to account for counter-current flow due to buoyancy as well as capillarity effects. In the following, we briefly describe the most commonly used ordering methods.

Cascade method

Appleyard and Cheshire (1982), proposed a cell-based ordering, the cascade method, to accelerate the standard Newton's method as follows. At each Newton iteration, we solve the linear system 1.24 for Newton updates, just like we do in standard Newton's method. Next, we update pressure variables (only) and compute phase potentials at each cell. We then order the cells based on phase potentials, from the highest potential to the lowest. We visit the cells based on the new order, and for each cell, we solve the two conservation equations (in the case of a two-phase flow system) simultaneously for pressure and saturation. The computed saturations are taken to be the saturation solution for the nonlinear iteration, but the pressure obtained from the local solution is only used for computing the influx for subsequent single-cell problems to ensure mass conservation. The pressure values obtained from the global linear system are taken to be the pressure solution for the nonlinear iteration.

Kwok (2007), proved that the cascade method converges to the solution in two iterations for an incompressible one-dimensional model problem in the absence of

counter-current flow. However, in the presence of counter-current flow or in multiple dimensions, convergence of the cascade method is not guaranteed.

Natvig's method

Natvig et al. (2006) and Natvig and Lie (2008), presented a family of solvers for hyperbolic transport equations in the absence of gravity and capillarity. The solvers are based on discontinuous Galerkin spatial discretization. They showed that the equations can be solved on a cell-by-cell basis by applying an optimal reordering of grid cells. By reordering the grid cells, the nonlinear system will have a lower triangular block structure, where each block corresponds to the degrees-of-freedom in a single or a small number of cells. The reordering algorithm is based on a directed graph with grid cells as vertices and the total velocity (flux) over cell interfaces as edges. They provide a topological sorting of the directed graph, allowing to visit the blocks in the direction of total velocity field sequentially, and solving for the degrees-of-freedom in each block, from upstream to downstream, using a standard Newton algorithm.

Phase-based potential ordering

Kwok and Tchelepi (2007), presented an ordering of equations and unknowns based on phase potentials. They showed that the saturation dependence in the Jacobian takes a lower-triangular form, allowing solving for saturation unknowns one variable at a time. By using different orderings corresponding to different phases, they extended the method to account for counter-current flow due to gravity. Based on the new phase-based potential ordering, they derived a reduced Newton algorithm for multiphase flow in porous media. They performed a rigorous mathematical analysis for one-dimensional problems and proved that in the absence of counter-current flow, the algorithm is always convergent. However, in the presence of counter-current flow,

reduced Newton converges only when a backward CFL condition is satisfied. When the backward CFL number is much larger than 1, it is possible for reduced Newton to cycle or diverge (Kwok (2007)).

1.4 Thesis overview

The main contribution of this thesis to the existing literature is a new ordering-based nonlinear solver that addresses the coupling between the flow and transport equations in an efficient and adaptive manner. At each FIM iteration, the new nonlinear solver allows us to take advantage of decoupled pressure and saturation updates, while saturation variables are updated on a cell-by-cell basis in the ‘loosely coupled’ regions. On the other hand, a simultaneous updating strategy is used for the ‘strongly connected’ regions which exhibit a tight coupling between pressure and saturation variables. We show that our proposed approach, along with the proposed Newton safeguard strategy, can efficiently handle challenging problems with counter-current flow due to gravity. The remainder of the thesis is organized as follows.

In Chapter 2, we introduce a safeguard strategy based on trust regions of flux function for problems with counter-current flow due to gravity. We use the algorithm to safeguard the saturation iterates in a fully coupled FIM solver. We show, by a variety of numerical examples, that the convergence behavior of FIM is significantly improved. Convergence is obtained for very large timesteps, a few orders of magnitude larger than what standard Newton’s method can handle.

In Chapter 3, we first review some of the key aspects of the nonlinear coupling between pressure and saturation. We then overview the phase-based potential ordering and revisit reduced Newton algorithm for challenging case of strong counter-current flow due to buoyancy. We present and discuss the convergence issues of reduced Newton algorithm.

In Chapter 4, we present the main contribution of the thesis. We propose a new nonlinear solver, based on which we no longer need to update the unknowns in a fully coupled fashion, everywhere and all the time. The crux of the algorithm comes in using a hybrid strategy to ensure that coupling is resolved in an adaptive yet efficient fashion by treating co-current and counter-current flow regions differently. We present a wide variety of examples in 2D and 3D and compare the computational efficiency of the proposed algorithm with the current state-of-the-art.

We present our conclusions and future directions in Chapter 5.

Chapter 2

Newton Safeguard Based on Flux Function

The fully implicit method (FIM) is widely used to solve the equations that govern multiphase flow in porous media. The nonlinear solver is typically based on Newton's method which is an iterative solution strategy. Convergence of Newton's method is generally not guaranteed for large time steps. It is desired to have a nonlinear solver that converges all the time regardless of timestep size. If the nonlinear solver is unconditionally convergent, timestep selection is solely based on accuracy considerations and is not restricted by the nonlinear solver itself. In the current state-of-the-art simulators, a timestep size is chosen and convergence of the nonlinear solver is checked at the end of each iteration. If convergence is not obtained after a certain number of iterations specified by the user, timestep is chopped and the nonlinear loop is repeated again. In this case, the iterations corresponding to the initial timestep are wasted and thus this strategy is computationally inefficient. Heuristic approaches are often used to improve convergence of Newton's method. These heuristics tend to safeguard Newton iterations such that change of specific variables over iterations does not exceed a certain value. One of the most widely used heuristics is the so-called

Appleyard chop method which targets saturation changes over nonlinear iterations on a cell-by-cell basis (Geoquest (2005)).

While heuristic approaches lead to a better convergence behavior for Newton's method, they are incapable of associating the existing nonlinearities with physics of the problem at hand (Geoquest (2005) and Naccache (1997)). For instance, in the case of multiphase flow, it can be shown that shape of fractional flow (flux) functions plays the key role in the nonlinear behavior of transport equations (Kwok (2007) and Jenny et al. (2009)). In this chapter, we revisit Newton's method and its convergence issues for large timestep. We then introduce a safeguard strategy for the fully implicit method (FIM) that addresses convergence issues due to inherent nonlinearity of flux functions. We show that the safeguarded nonlinear solver is capable of resolving timestep size limitations for the most challenging nonlinear problems with special focus on counter-current flow due to buoyancy.

2.1 Nonlinearity of transport equation

In order to have a better understanding of the underlying nonlinearities, we start off by analyzing transport equation in its simplest form. Suppose we have a single-cell displacement problem where we inject water from the top into a medium initially filled with oil. The injection boundary condition is set to unit water saturation at the injection interface ($S_{inj} = 1$). For this scenario, the transport equation for water is given by

$$\frac{\partial S_w}{\partial t} + \frac{\partial (f_w u)}{\partial z} = 0, \quad (2.1)$$

where S_w , f_w and u denote water saturation, water fractional flow (flux) function and total velocity respectively. An implicit finite-difference discretization of Equation 2.1 subject to initial and boundary conditions, results in the residual equation in the

following form

$$\frac{S_w^{n+1}}{c} + (f_w - 1) = 0, \quad (2.2)$$

where $c = \frac{u\Delta t}{\Delta z}$ is the Courant number (Δt and Δz denote timestep size and grid size). As Equation 2.2 shows, the residual consists of two terms which behave differently as far as nonlinearities are concerned (Younis (2011)). The first term, $\frac{S_w^{n+1}}{c}$ is linear with respect to saturation, whereas the second term $(f_w - 1)$ can have a linear or nonlinear shape depending on the fractional flow function. The overall nonlinear behavior of the residual depends on the Courant number which determines the relative contribution of these two terms. Figure 2.1 shows plots of the discrete residual of the transport equation versus water saturation for different Courant numbers. The residual function tends to be more linearly shaped for smaller Courant numbers. In the limit, when $c \rightarrow 0$, the residual turns into a linear function. On the other hand, for large values of c , more nonlinearity is observed and therefore it takes a larger number of Newton iterations for convergence to the solution. In the limit, when $c \rightarrow \infty$, residual turns into $f_w - 1$, which implies that shape of flux function directly affects nonlinear behavior of residual function. This example shows that transport equation has a more complicated nonlinear behavior when large timesteps (leading to large Courant numbers) are chosen and shape of the nonlinearity of fractional flow function plays the key role in nonlinear behavior of residual.

2.2 Convergence maps

In the previous section, we observed that fractional flow (flux) functions play the key role in nonlinear behavior of residual function corresponding to transport equation. In order to better understand such nonlinearities, we focus on flux functions and investigate the relationship between shape of flux functions and convergence behavior of Newton's method. We try four different types of flux functions: concave, convex,

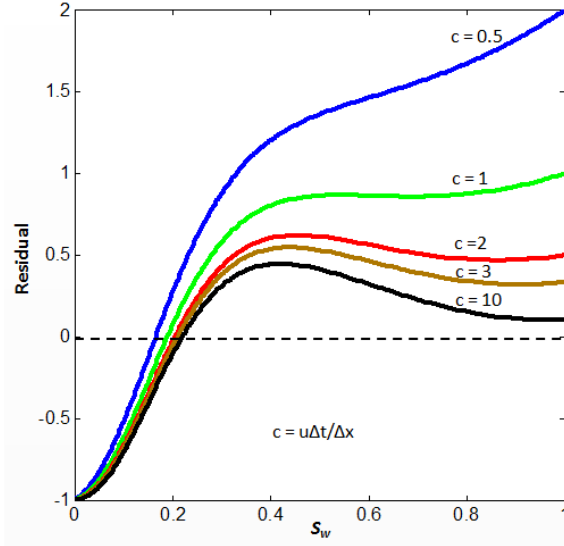


Figure 2.1: Effect of Courant number on shape of residual function corresponding to single-cell water injection problem. The intersection of the dashed line with each residual function represents the solution. The figure shows that for large timesteps (large Courant numbers), nonlinear behavior of residual function is influenced by shape of fractional flow function.

S-shaped and mixed (gravity flux) which are typical flux functions for transport in porous media. For each function, we solve the nonlinear equation $f_w(S_w) = f_w^*$, where f_w^* is a target flux. For a given target flux, we choose different initial guesses, ranging from $S_w^{(0)} = 0$ to $S_w^{(0)} = 1$. We repeat this process for different target fluxes ranging from $f_w^* = 0$ to $f_w^* = f_w^{max}$. For each $(f_w^* - S_w^{(0)})$ pair, we record the number of iterations required for convergence of Newton's method. The number of iterations are visualized using different colors in the $(f_w^* - S_w^{(0)})$ plane. The resulting convergence map provides a great deal of information about nonlinearity of different flux functions.

2.2.1 Concave flux

Figure 2.2 (left) shows a typical concave flux obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 0.1$. The convergence map corresponding to this flux function, presented in

Figure 2.2 (right), shows that Newton's method converges in at most 8 iterations for any target flux using any arbitrary initial guess. Repeating the exercise for different combinations of relative permeability and viscosity ratios leading to different concave fluxes, implies that Newton's method is unconditionally convergent for this type of flux functions.

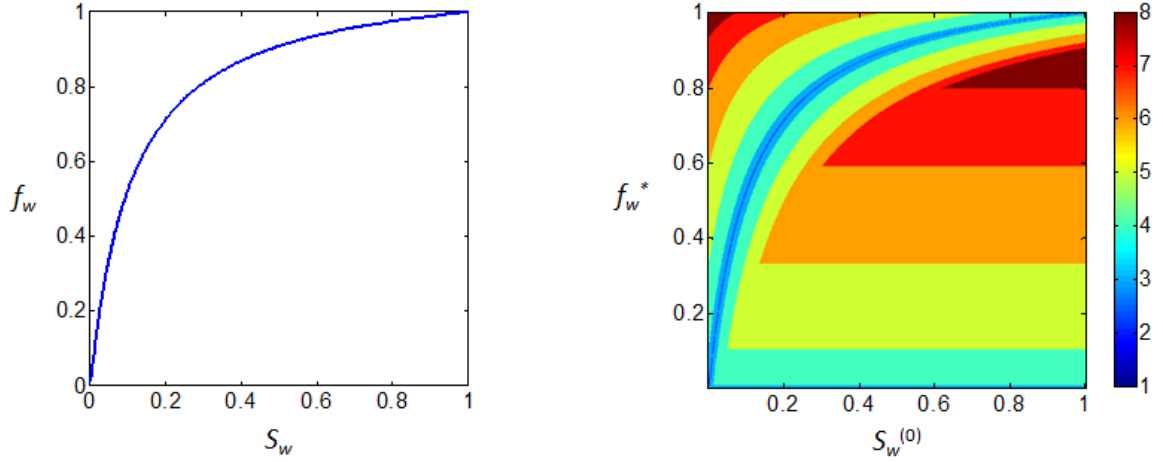


Figure 2.2: A typical concave flux function shown on the left, obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 0.1$. The corresponding convergence map is shown on the right. Newton's method is unconditionally convergent in the case of concave flux functions.

2.2.2 Convex flux

Figure 2.3 (left) shows a typical convex flux obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 10$. The convergence map corresponding to this flux function, presented in Figure 2.3 (right), shows that similar to the previous case, Newton's method converges in at most 8 iterations for any target flux using any arbitrary initial guess. Repeating the exercise for different combinations of relative permeability and viscosity ratios leading to different convex fluxes, implies that Newton's method is unconditionally convergent in the case of convex fractional flow functions.

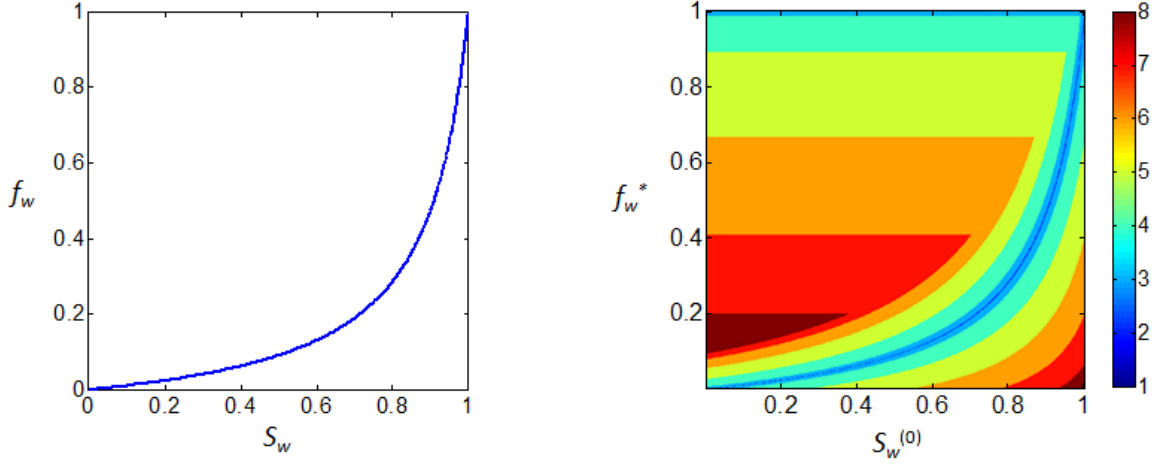


Figure 2.3: A typical convex flux function shown on the left, obtained using $k_{rw} = S_w$, $k_{ro} = 1 - S_w$ and $\frac{\mu_w}{\mu_o} = 10$. The corresponding convergence map is shown on the right. Newton's method is unconditionally convergent in the case of convex flux functions.

2.2.3 S-shaped flux

Figure 2.4 (left) shows a S-shaped flux obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$ and $\frac{\mu_w}{\mu_o} = 10$. S-shaped flux functions, as opposed to concave and convex fluxes, have an inflection point over which the sign of the second derivative of flux with respect to saturation changes. The convergence map corresponding to this flux function, presented in Figure 2.4 (right), shows that Newton's method does not converge for all pairs of target flux and initial guess. The white areas in the map correspond to the pairs for which convergence is not obtained in 30 iterations or fewer. Repeating this exercise for other viscosity ratios confirms that Newton's method converges only for certain combinations of target flux and initial guess, see Figure 2.5 corresponding to $\frac{\mu_w}{\mu_o} = 10$. Notice that for S-shaped fluxes, convergence of Newton's method is always guaranteed if the inflection point is chosen as the initial guess.

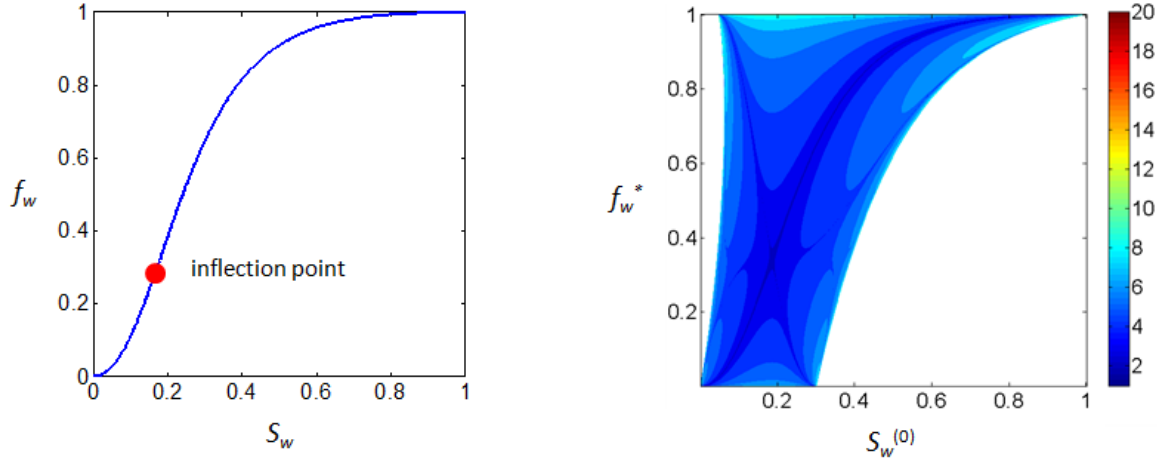


Figure 2.4: A typical S-shaped flux function shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$ and $\frac{\mu_w}{\mu_o} = 0.1$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas; however convergence is always guaranteed if inflection point is chosen as the initial guess.

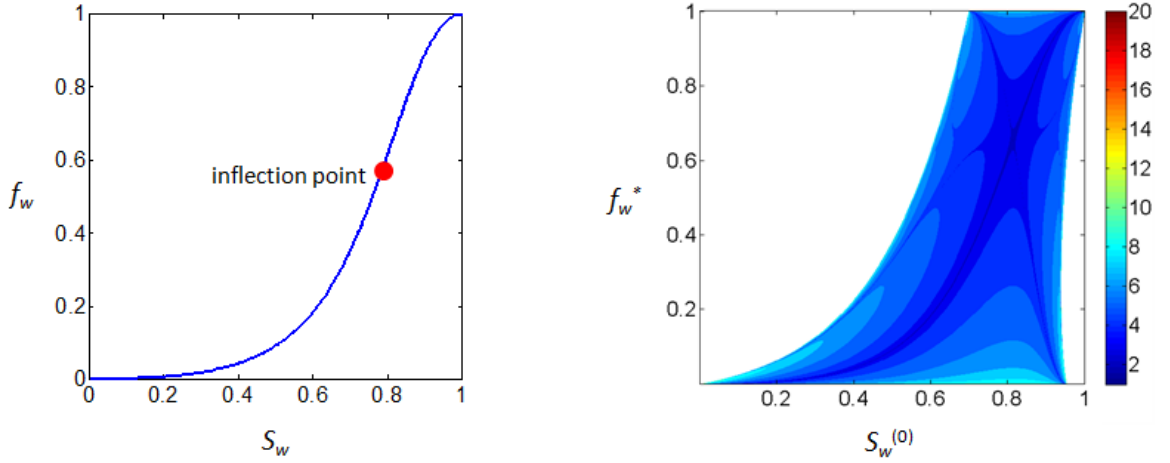


Figure 2.5: A typical S-shaped flux function shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$ and $\frac{\mu_w}{\mu_o} = 10$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas; however convergence is always guaranteed if inflection point is chosen as the initial guess.

2.2.4 Flux in the presence of gravity

Figure 2.6 (left) presents a typical flux function in the presence of gravity obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$, $\frac{\mu_w}{\mu_o} = 0.1$ and $N_g = -2$. The shape of flux function in this case is more complicated than concave, convex or S-shaped fluxes. The function has a sonic point, which corresponds to the point at which the first derivative of flux function with respect to saturation is zero. An inflection point exists on each side of the sonic point over which the second derivative of the flux function with respect to saturation changes sign. Figure 2.6 (right) shows the corresponding convergence map. The white areas in the convergence map indicate that Newton's method does not converge for all pairs of target flux and initial guess for the nonlinear problem based on flux in the presence of gravity.

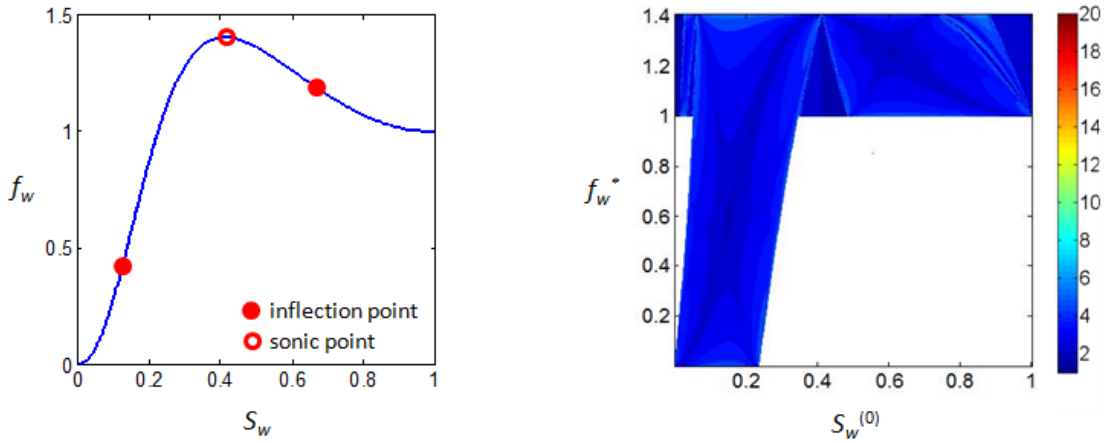


Figure 2.6: A typical flux function in the presence of gravity shown on the left, obtained using $k_{rw} = S_w^2$, $k_{ro} = (1 - S_w)^2$, $\frac{\mu_w}{\mu_o} = 0.1$ and $N_g = -2$. The corresponding convergence map is shown on the right. Newton's method is not convergent in the white areas of the convergence map.

2.3 Newton safeguard based on trust regions of flux function

It was shown that convergence of Newton's method is not guaranteed for solving the nonlinear problem in the form of $f_w(S_w) = f_w^*$ when f_w is a S-shaped or a gravity flux function. In order to guarantee convergence, we need to use safeguard strategies that limit Newton iterates such that they eventually converge to the solution. In this section, we first revisit the safeguard strategy for S-shaped flux functions proposed by Jenny et al. (2009) which is based on flux functions. We then extend the safeguard algorithm to account for counter-current flow due to buoyancy which leads to more complicated flux functions in the presence of gravity.

2.3.1 Safeguard for S-shaped flux function

Jenny et al. (2009) introduced a Newton safeguard for S-shaped flux functions based on trust regions of the flux function. The main idea is that convergence of Newton's method is guaranteed as long as Newton iterates lie on one side of the inflection point of flux function. This implies that convergence is always obtained if during Newton iterations, the iterates are set back to the inflection point each time they cross it. In other words, if the second derivatives of flux evaluated at the old and new iteration levels have different signs, i.e. $f''(S^{\nu+1})f''(S^\nu) < 0$, the new iterate is set to: $S^{\nu+1} = S_{inf} \pm \epsilon$ where S_{inf} is the saturation corresponding to the inflection point and ϵ is a small number used for numerical stability. Figure 2.7 shows convergence maps obtained using this safeguard strategy for S-shaped flux functions presented in Figures 2.4 (left) and 2.5 (left). These maps show that Newton's method with safeguard converges unconditionally in the entire $(f_w^* - S_w^{(0)})$ plane.

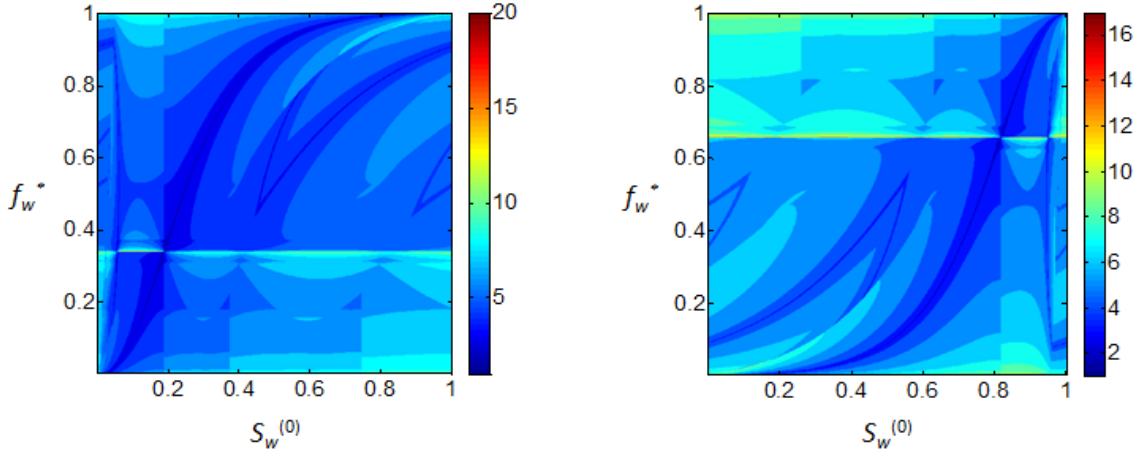


Figure 2.7: Convergence maps for S-shaped flux functions (presented in Figures 2.4 and 2.5), obtained using the Newton safeguard. Convergence is obtained everywhere in the $(f_w^* - S_w^{(0)})$ plane.

2.3.2 Extension of the safeguard algorithm for flux functions with gravity

In the case of counter-current flow due to gravity, flux functions are no longer S-shaped. In order to extend the existing safeguard algorithm for S-shaped functions to the more general case of flux in the presence of gravity, we propose the following algorithm: We divide the flux function into two regions separated by the sonic point. Each of these regions has an inflection point. In order to safeguard Newton iterates, we first check if the old and new iterates are on different sides of the sonic point by checking if $f'(S^{\nu+1})f'(S^\nu) < 0$. If this statement is true, the iterates have crossed the sonic point which is an indication of traveling from one region to another. In this case, we set the new iterate back the sonic point, i.e. $S^{\nu+1} = S_{son} \pm \epsilon$, where S_{son} is the saturation corresponding to the sonic point. On the other hand, if the old and new iterates both lie within the same region, we check if the corresponding inflection point is crossed (similar to the case of S-shaped flux), in which case we set the iterate back to

the inflection point, i.e. $S^{\nu+1} = S_{inf} \pm \epsilon$, where S_{inf} is the saturation corresponding to the inflection point (see Algorithm 1). The convergence map obtained using Newton's method with the proposed safeguard algorithm is presented in Figure 2.8. The figure shows that Newton's method is convergent for all possible combinations of target flux and initial guess.

Algorithm 1 Newton safeguard based on trust regions of flux function.

```

1: procedure SAFEGUARD ( $S^\nu, S^{\nu+1}, N_g$ )
2:   if  $f'(S^{\nu+1}, N_g) f'(S^\nu, N_g) < 0$  then
3:     if  $S^{\nu+1} > S^\nu$  then
4:        $S^{\nu+1} = S_{son} + \epsilon$  ;
5:       go to 10
6:     else
7:        $S^{\nu+1} = S_{son} - \epsilon$  ;
8:       go to 20
9:     end if
10:  end if
11:  if  $f''(S^{\nu+1}, N_g) f''(S^\nu, N_g) < 0$  then
12:    if  $S^{\nu+1} > S^\nu$  then
13:       $S^{\nu+1} = S_{inf} + \epsilon$  ;
14:      go to 20
15:    else
16:       $S^{\nu+1} = S_{inf} - \epsilon$  ;
17:      go to 20
18:    end if
19:  end if
20: end procedure

```

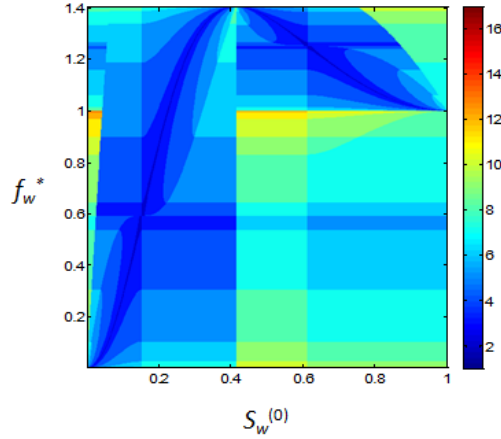


Figure 2.8: Convergence map for flux function with gravity (presented in Figure 2.6), obtained using the proposed Newton safeguard. Convergence is obtained everywhere in the $(f_w^* - S_w^{(0)})$ plane.

2.4 The fully coupled FIM with Newton safeguard

In the previous sections, we focused mainly on the underlying nonlinearities of flux functions. We proposed a safeguard algorithm that results in convergence of Newton's method for a simple nonlinear problem that aims to solve for a target flux given an initial guess for saturation. In reality, the nonlinear equations that govern multiphase flow and transport are more involved; however the underlying nonlinearity is mostly dominated by flux functions (Kwok (2007)). Our main objective here is to employ the flux function-based Newton safeguard for solving gravity-dominated two-phase flow and transport equations in a fully implicit setting where pressure and saturations are solved for simultaneously. The main idea is to safeguard saturation iterates for each grid cell based on the flux functions at the cell interfaces after each Newton iteration. Without loss of generality, we describe the algorithm for a 2-D structured grid in $x-z$ plane where gravity acts in z direction.

Algorithm 2 shows the pseudocode describing the fully coupled FIM with Newton

safeguard for solving two-phase flow and transport equations. In the beginning of each Newton iteration, given the saturation and pressure at the old iteration level, ν , we first compute the total velocity by summing up the individual phase velocities. Next, we compute the flux functions at cell interfaces, both in x and z directions, denoted by f_h and f_v respectively. We use the total velocities to compute the gravity numbers for interfaces in the z direction, which are used for computation of f_v . Notice that, since there is no dependence of f_h on the gravity number, f_h is the same for all interfaces in the x direction; however, in the presence of gravity, f_v varies across interfaces. This is because we have different gravity numbers due to the difference in total velocities at interfaces in z direction. In the next step, we update the pressure and saturation of grid cells by solving the linearized system of equations obtained from the corresponding Jacobian matrix and residual vector computed at the old iteration level. In order to safeguard the newly computed saturation iterates, we first ensure that they are in the right physical range by limiting the saturations between 0 and 1. At this point, we need to check whether the iterates cross the inflection points and the sonic point of the corresponding flux functions. For each grid cell, we first consider f_h corresponding to the interface in x direction (which is the same function for all interfaces in x direction). If the old and new saturation iterates lie on different sides of the inflection point, we set the iterate back such that it barely crosses the inflection point. Next, we consider f_v corresponding to a vertical interface. If the new and old saturation iterates are on different sides of the sonic point of f_v , we set the iterate back such that it barely crosses the sonic point. On the other hand, if the sonic point is not crossed, it means that iterates are staying in the same region of flux function, in which case we only check whether the inflection point corresponding to the region is crossed. If that is the case, we set the iterate back such that the inflection point is barely crossed. The saturations at the end of this step are considered the final iterates for the current Newton iteration. Newton iterations are repeated until

convergence is obtained.

Algorithm 2 Algorithm describing the fully coupled FIM with Newton safeguard based on trust regions of flux function.

```

1: while convergence criterion is not met do
2:   Compute  $N_g^\nu$  at all cell interfaces ;
3:   Compute Jacobian  $\mathbf{J}^\nu$ , and residual  $\mathbf{r}^\nu$  ;
4:   Solve for Newton update from  $\mathbf{J}^\nu \delta \mathbf{x}^\nu = -\mathbf{r}^\nu$  ;
5:   Compute  $\mathbf{p}^{\nu+1} = \mathbf{p}^\nu + \delta \mathbf{p}^\nu$  ;
6:   Compute  $\mathbf{S}^{\nu+1} = \mathbf{S}^\nu + \delta \mathbf{S}^\nu$  ;
7:   for  $i = 1 \rightarrow N_{cells}$  do
8:      $S_i^{\nu+1} = \max(S_i^{\nu+1}, 0)$  ;
9:      $S_i^{\nu+1} = \min(S_i^{\nu+1}, 1)$  ;
10:    SAFEGUARD  $(S_i^\nu, S_i^{\nu+1}, N_g^\nu)$  ;
11:   end for
12:    $\nu := \nu + 1$  ;
13: end while
```

2.5 Numerical examples

The following examples show the application of the proposed nonlinear solver for solving a variety of numerically challenging problems. The examples cover a wide range of cases with special focus on counter-current flow due to buoyancy. We compare convergence behavior of the new nonlinear solver with the current state-of-the-art.

2.5.1 Example 1: Top-down gravity segregation with injection from the top (1-D)

In the first example, we consider a homogeneous 1-D domain positioned vertically in z direction. The reservoir is discretized using $1 \times 1 \times 100$ cells in x , y and z directions respectively. Initially, the top half of the domain is filled with a heavy fluid (water) resting on top of a light fluid (oil) filling the lower half of the domain. The initial pressure is 150 psi everywhere. The rock and fluid properties (both incompressible) are listed in Table 2.1. Water is injected from the top at a constant rate of $q = 0.1$ bbl/day and fluids are produced from the lower end at the same rate. The pressure at the producing grid cell is set to 100 psi. We use the fully coupled FIM with the proposed flux function-based Newton safeguard to solve the flow and transport equations. Figure 2.9 shows the saturation profile at $t = 10$ cell PVI. The simulation was run using only one large timestep, $\Delta t = 10$ cell PVI. The variation of CFL number in z direction at the final simulation time is presented in Figure 2.10. The number of iterations required for convergence of the nonlinear solver for different timesteps is shown in Figure 2.11. The dashed line shows that standard Newton's method without any safeguard, converges only for small timesteps ($\Delta t \leq 1$ cell PVI), comparable to allowable timesteps using an IMPES scheme; however convergence is guaranteed for very large timesteps (only presented for timesteps as large as $\Delta t = 30$ cell PVI in the figure) when Newton iterations are safeguarded based on trust regions

of flux function.

Property	Value
K (mD)	100
ϕ	0.2
ρ_w ($lb.ft^{-3}$)	100
ρ_o ($lb.ft^{-3}$)	50
μ_w (cP)	0.2
μ_o (cP)	1

Table 2.1: Rock and fluid properties used in example 1.

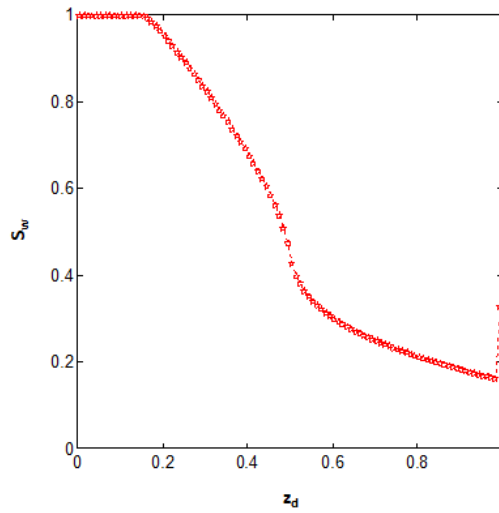


Figure 2.9: Saturation profile at $t = 10$ cell PVI, corresponding to example 1, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 10$ cell PVI.

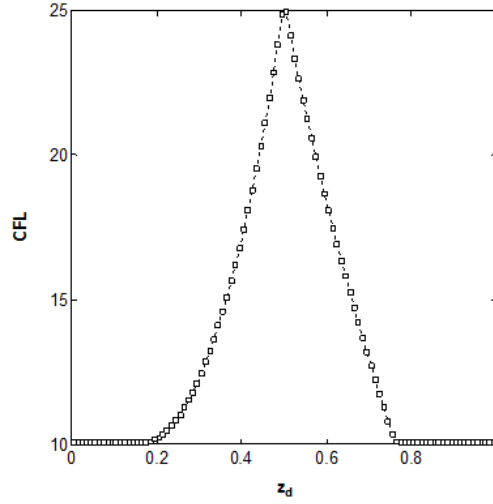


Figure 2.10: The CFL number versus dimensionless height, at $t = 10$ cell PVI, corresponding to example 1, obtained using the fully coupled FIM with the flux function-based Newton safeguard.

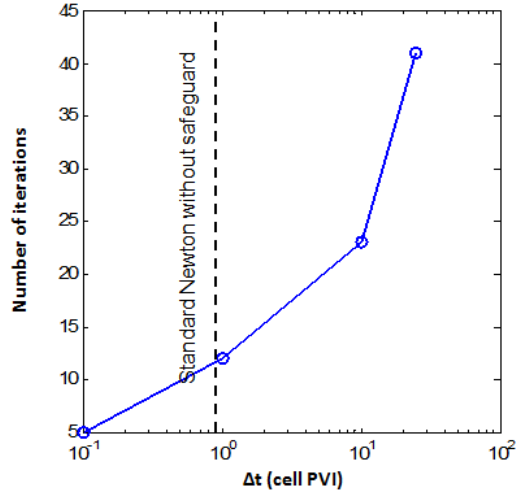


Figure 2.11: Number of iterations versus timestep size corresponding to example 1. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 1$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

2.5.2 Example 2: Homogeneous quarter 5-spot with gravity

Consider a homogeneous quarter 5-spot. The reservoir is discretized using $50 \times 1 \times 50$ cells in x , y and z directions respectively. The rock and fluid properties are similar to those used for the first example presented in Table 2.1. Initially, the medium is filled with oil and pressure is 150 psi everywhere. Water is injected from the upper left corner at the constant rate of $q = 0.1$ bbl/day and fluids are produced from the lower right corner at the same rate. Pressure at the producing grid cell is set to 100 psi. The total simulation time is $t = 1250$ cell PVI and only one large timestep, $\Delta t = 1250$ cell PVI, is used for the entire simulation. Figure 2.12 presents pressure map (left) and saturation map (right) at the final simulation time. Figure 2.13 shows the CFL numbers at various locations at the final simulation time. Notice that the color bar for the CFL map is clipped at an upper bound of 200. The maximum throughput achieved in this simulation is 1250 cell PVI which is well above the maximum throughput obtained using IMPES or standard Newton solvers. Figure 2.14 shows the plot of number of Newton iterations versus timestep size. As shown by the dashed line, standard Newton without the flux function-based safeguard does not converge for $\Delta t \geq 2$ cell PVI; however the nonlinear solver converges for much larger timesteps when Newton iterates are safeguarded based on the proposed algorithm.

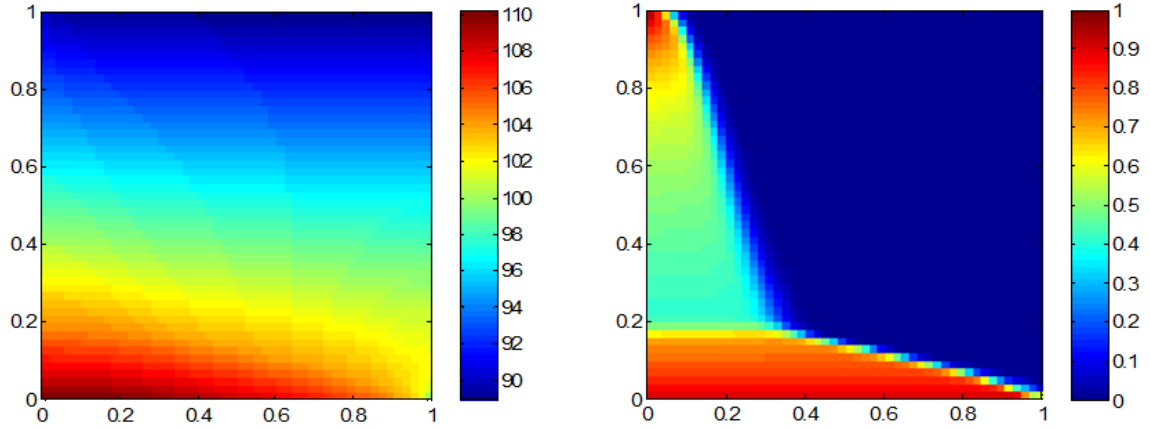


Figure 2.12: Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 2, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.

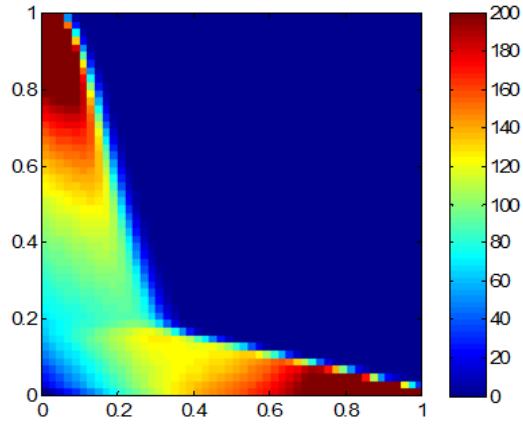


Figure 2.13: The CFL map, at $t = 1250$ cell PVI corresponding to example 2, obtained using the fully coupled FIM with the flux function-based Newton safeguard.

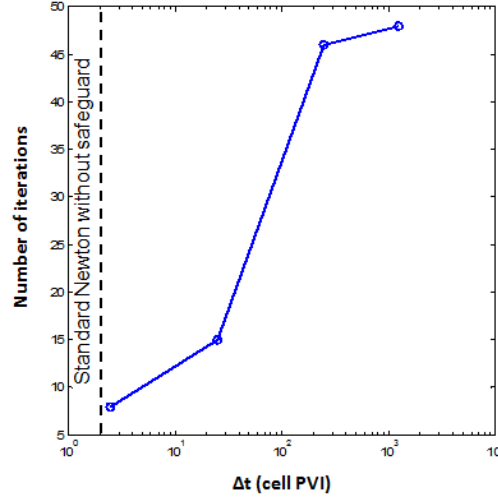


Figure 2.14: Number of iterations versus timestep size corresponding to example 2. The dashed line shows that standard Newton’s method without the proposed safeguard, does not converge for $\Delta t \geq 1$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

2.5.3 Example 3: Top-down gravity segregation with injection in a homogeneous quarter 5-spot

The geometry of the domain, the rock and fluid properties and injection/production conditions are identical to those of example 2. Initially, the top half of the domain is filled with water and the lower half is filled with oil. Initial pressure is 150 psi everywhere. The total simulation time is $t = 1250$ cell PVI and only one large timestep, $\Delta t = 1250$ cell PVI, is used for the entire simulation. The pressure and saturation maps at the final simulation time are presented in Figure 2.15. The saturation map clearly shows the redistribution of light and heavy due to buoyancy. The map of CFL number with the color bar clipped at an upper bound of 200 is shown in Figure 2.16. The maximum throughput achieved in this case is 1250 cell PVI. Figure 2.17 shows the plot of number of Newton iterations versus timestep size. The dashed line on the figure indicates the maximum timestep size for which convergence is obtained using

standard Newton without the flux function-based safeguard ($\Delta t = 20$ cell PVI). The nonlinear solver with the proposed Newton safeguard converges for timesteps much larger than standard Newton can handle.

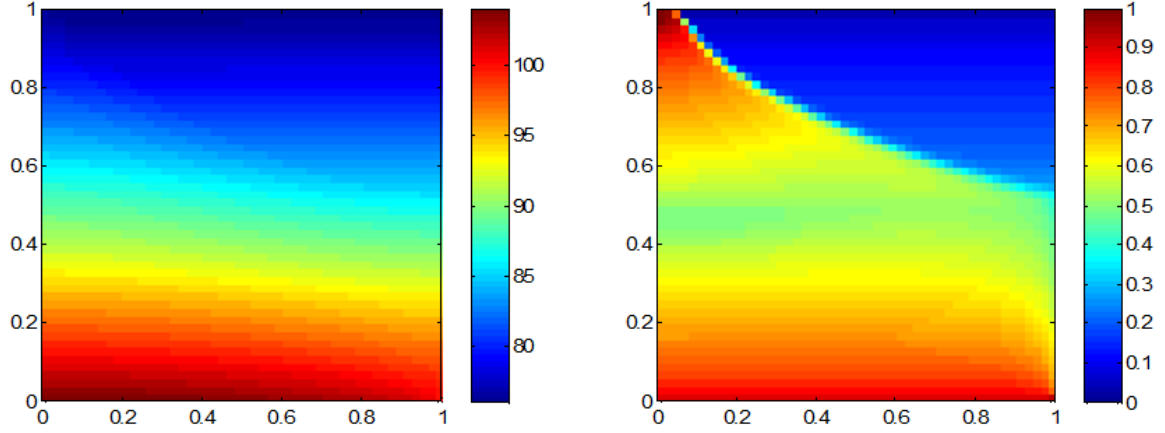


Figure 2.15: Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 3, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.

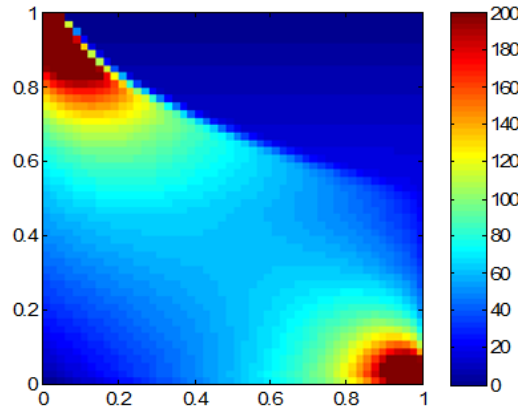


Figure 2.16: The CFL map, at $t = 1250$ cell PVI corresponding to example 3, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.

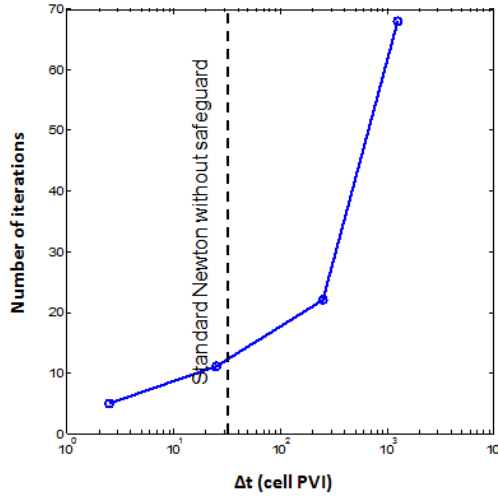


Figure 2.17: Number of iterations versus timestep size corresponding to example 3. The dashed line shows that standard Newton’s method without the proposed safeguard, does not converge for $\Delta t \geq 30$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

2.5.4 Example 4: Lock-exchange with injection in a homogeneous quarter 5-spot

Similar to the previous two cases, we consider a quarter 5-spot with the same rock and fluid properties. Initially, the left half of the domain is filled with water and the right half is filled with oil. Initial pressure is 150 psi everywhere. The injection and production conditions as well as final simulation time and timestep size are also identical to previous examples. Figures 2.18 and 2.19 present maps of pressure and saturation as well as CFL number at the final simulation time while the color bar for the CFL map is clipped at an upper bound of 200. Figure 2.20 shows that maximum allowable timestep size for standard Newton is much smaller than what the proposed nonlinear solver can aim for.

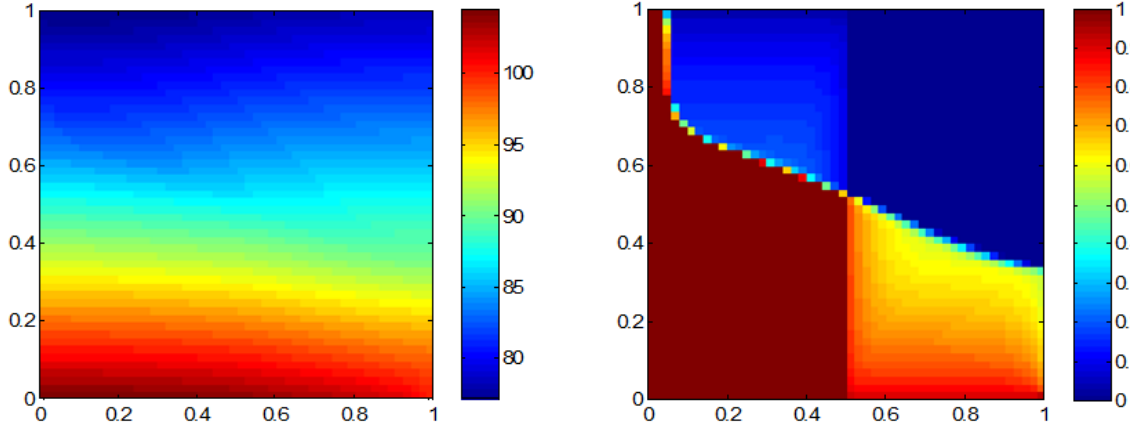


Figure 2.18: Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 4, obtained using the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.

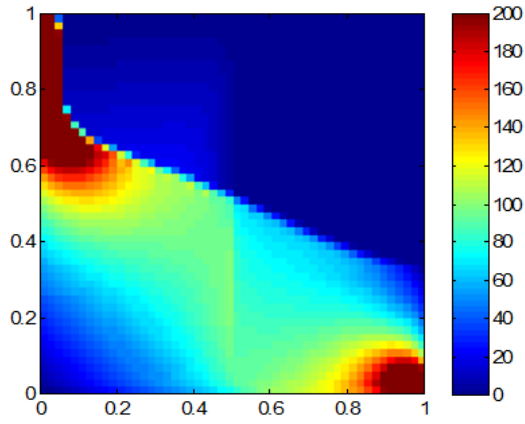


Figure 2.19: The CFL map, at $t = 1250$ cell PVI corresponding to example 4, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.

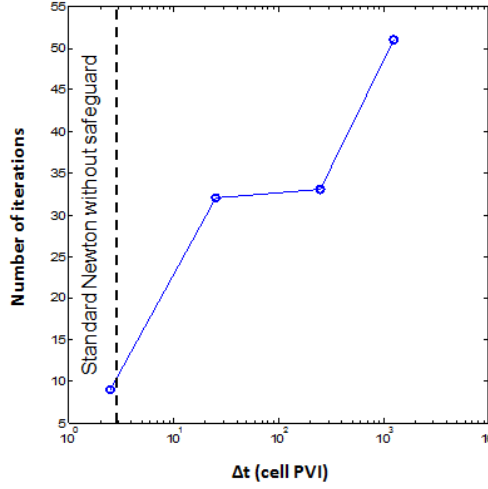


Figure 2.20: Number of iterations versus timestep size corresponding to example 4. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t \geq 3$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

2.5.5 Example 5: Top-down gravity segregation with injection in a heterogeneous quarter 5-spot

In order to test the capability of the nonlinear solver with Newton safeguard for solving heterogeneous problems, we add significant heterogeneity to the quarter 5-spot model. Figure 4.10 shows the log-normal permeability field used in this example with four orders of magnitude difference in permeability values. The initial saturation condition corresponds to the top-down gravity segregation identical to example 3. Rock and fluid properties, injection/production conditions and timestep size are chosen identical to the previous three examples. Figures 2.22 and 2.23 show maps of pressure and saturation as well as CFL number at the final simulation time while the color bar for the CFL map is clipped at an upper bound of 100. Figure 2.24 shows that even in the case of significant heterogeneity, the proposed nonlinear solver still allows picking much larger timesteps as compared to standard Newton.

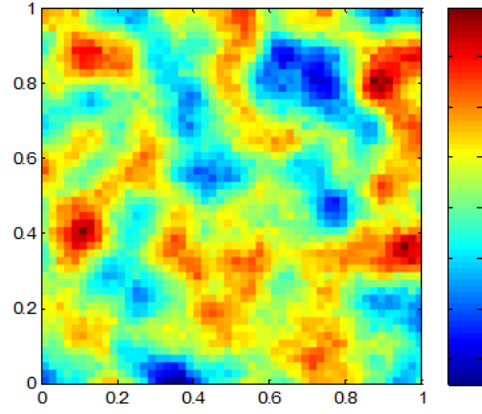


Figure 2.21: Log-normal permeability field used for heterogeneous examples.

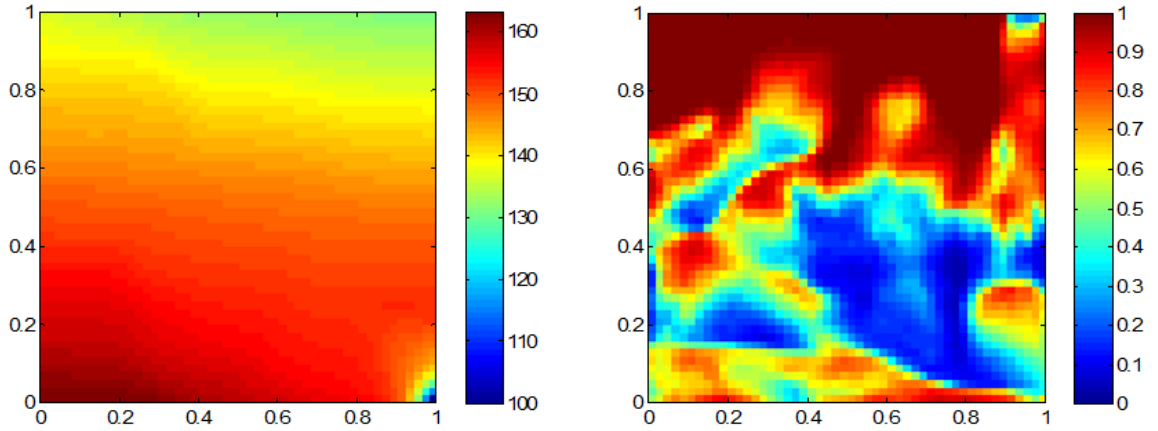


Figure 2.22: Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 5, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.

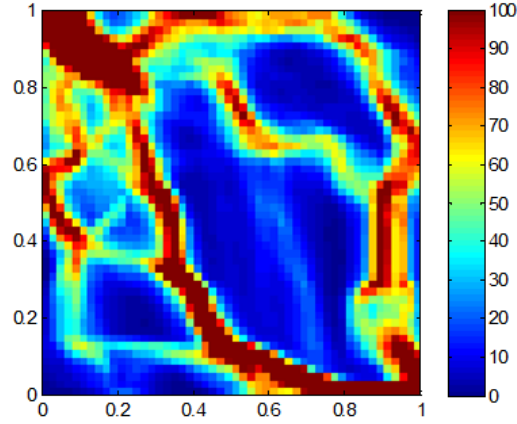


Figure 2.23: The CFL map, at $t = 1250$ cell PVI corresponding to example 5, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.

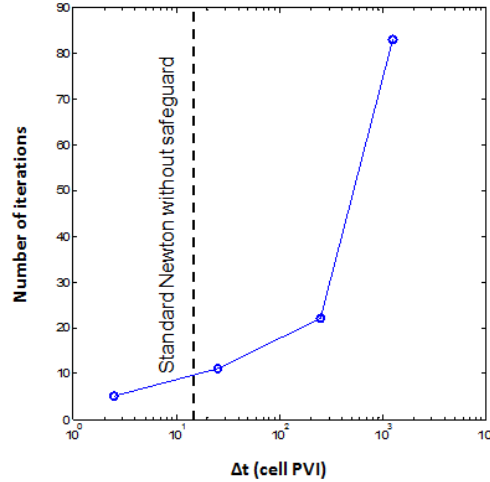


Figure 2.24: Number of iterations versus timestep size corresponding to example 5. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t > 15$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

2.5.6 Example 6: Lock-exchange with injection in a heterogeneous quarter 5-spot

For the last example, we solve the lock-exchange problem (as described in example 4) on a heterogeneous quarter 5-spot medium with permeability field presented in Figure 4.10. Similar to previous 2-D cases, the simulation is run for only one timestep using $\Delta t = 1250$ cell PVI. Pressure, saturation and CFL maps at the end of simulation time are shown in Figures 2.25 and 2.26 respectively. Figure 2.27 shows the number of Newton iterations versus timestep size. The the fully coupled FIM with flux function-based safeguard exhibits a superior convergence behavior as compared to standard Newton.

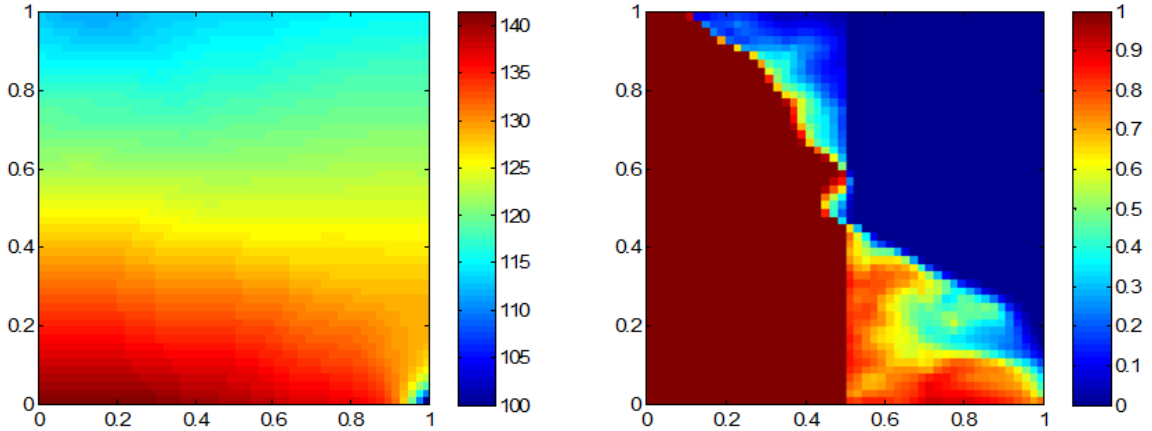


Figure 2.25: Pressure map (left) and saturation map (right) at $t = 1250$ cell PVI corresponding to example 6, obtained using the the fully coupled FIM with the flux function-based Newton safeguard. The simulation was run using only one timestep, $\Delta t = 1250$ cell PVI.

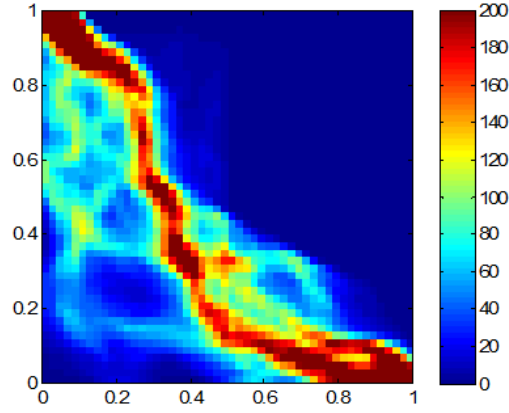


Figure 2.26: The CFL map, at $t = 1250$ cell PVI corresponding to example 4, obtained using the the fully coupled FIM with the flux function-based Newton safeguard.

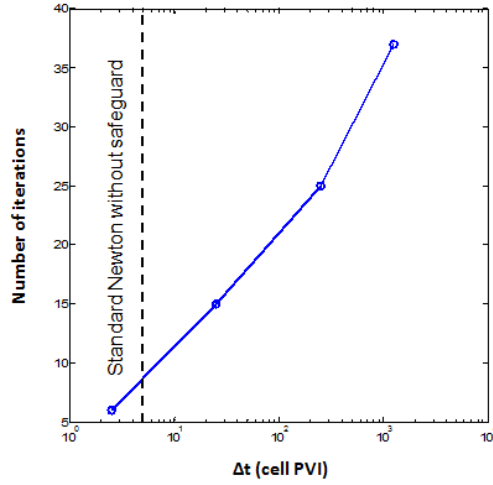


Figure 2.27: Number of iterations versus timestep size corresponding to example 6. The dashed line shows that standard Newton's method without the proposed safeguard, does not converge for $\Delta t > 5$ cell PVI. Using the flux function-based safeguard algorithm, unconditional convergence is obtained.

Chapter 3

Nonlinear Coupling of Flow and Transport

The main theme of this thesis is to design a nonlinear solver that can resolve the coupling between flow and transport equations in a more efficient manner than preserving the full coupling of variables in the FIM. The first step in tackling this problem is to gain insight into some of the key aspects of the nonlinear coupling. We start by comparing the convergence behavior of pressure and saturation, which leads to the idea of updating pressure in a selective manner. We then introduce two selective pressure updating strategies within the framework of fully coupled FIM. We investigate the effect of these two particular strategies on the convergence properties of the nonlinear solver.

In the remainder of the chapter, we focus on phase-based potential ordering (Kwok (2007)), which allows a partial decoupling of the problem into a sequence of single-cell problems. We review reduced Newton algorithm which applies Newton's method to a reduced system for pressure update and the single-cell nonlinear transport equations obtained from phase-based ordering of equations. We discuss the convergence issues of reduced Newton method for problems with tight coupling between flow and transport

due to strong counter-current flow and buoyancy effects. The materials presented in this chapter are the building block for our most efficient coupling strategy presented in the next chapter.

3.1 Convergence of pressure and saturation in the fully coupled FIM

The nonlinear equations governing flow and transport in porous media are coupled. In order to improve the convergence behavior of the nonlinear solvers that aim to solve these equations, we need to have a thorough understanding of the nature of the underlying coupling and the degree to which these equations depend on each other through the coupling terms. Our main focus here is to identify whether or not the coupled pressure and saturation variables converge to the solution with the same rate. It is also critical to understand how coupling of pressure and saturation variables, or equivalently flow and transport equations, impacts the convergence behavior of the fully coupled FIM.

To elaborate on the convergence behavior of the fully coupled FIM, we consider the numerical example outlined in section 2.7.3, discretized using $2 \times 1 \times 2$ cells in x , y and z directions respectively. We consider one large timestep size of $\Delta t = 80$ cell PVI. For grid cell i and at Newton iteration ν , we define the relative changes of pressure and saturation over Newton iteration ν as

$$\begin{aligned} \delta^* p_i^\nu &= \left| \frac{p_i^{\nu+1} - p_i^\nu}{p_i^\nu} \right|, \\ \delta S_{w,i}^\nu &= |S_{w,i}^{\nu+1} - S_{w,i}^\nu|, \quad \nu = 0, 1, \dots \end{aligned} \tag{3.1}$$

Figure 3.1 shows the maximum relative change of pressure and change of saturation over grid cells, with respect to the Newton iteration number. As the figure reveals,

pressure iterates tend to converge to the solution faster than saturation iterates. The pressure iterates, which are associated with alliptic PDEs, change significantly in the first few iterations and change slightly thereafter. The exact definition of ‘the first few iterations’ is problem dependent, but in our experience this number does not usually exceed 20% of the total number of iterations. On the other hand, the saturation iterates, which are associated with hyperbolic PDEs, show an entirely different behavior in the sense that saturation variations are not necessarily limited to the first few iterations.

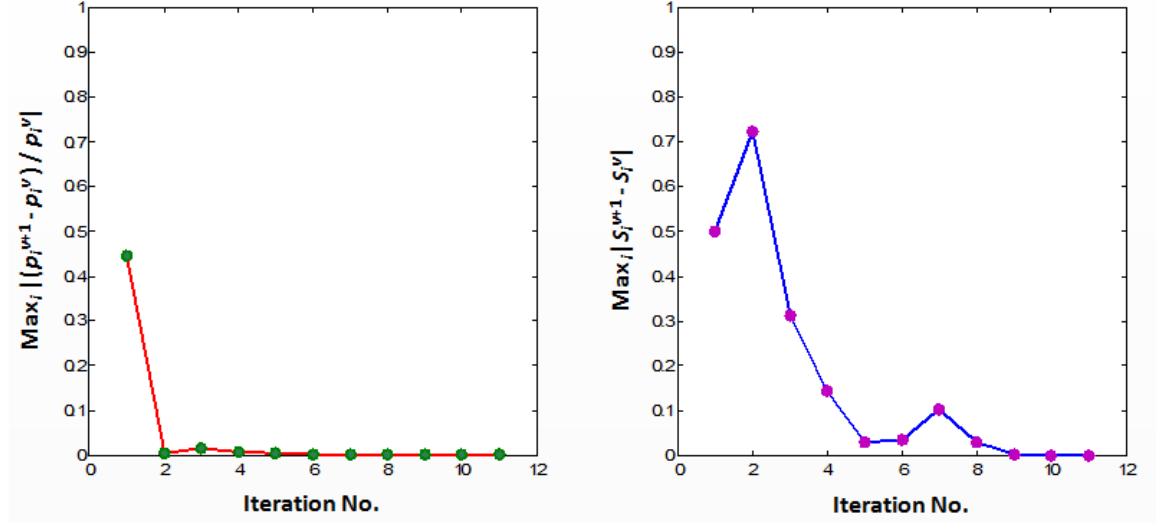


Figure 3.1: Maximum change of pressure iterates (left) and saturation iterates (right) over Newton iteration for the small test problem. The pressure iterates (corresponding to elliptic flow equation) tend to converge to solution faster than saturation iterates (corresponding to hyperbolic transport equation).

3.2 Effect of selective pressure updating on convergence of the fully coupled FIM

The difference between the convergence behaviors of pressure and saturation, leads to the idea of selective pressure updating. From a spatial standpoint, updates could

potentially take place only in selected parts of the domain as opposed to on every single grid cell. On the other hand, from a temporal point of view, selective pressure updating refers to updating pressure only in selected Newton iterations or selected timesteps, as opposed to updating all the time. In the following sections, we introduce two selective updating strategies and investigate their impact on convergence of the nonlinear solver.

3.2.1 Updating pressure only in the first few Newton iterations

The first proposed strategy for selective pressure update reads as follows. During the course of Newton iterations, update pressure iterates only in the first few iterations. For the subsequent iterations, discard pressure changes obtained from solution of the linear system. The hypothesis here is that it might not be necessary to update pressure after the first few iterations since pressure iterates essentially converge (or at least do not change significantly) thereafter. Notice that since we solve the linear system of equations at each Newton iteration, the algorithm proposed here does not really improve the computational efficiency of the FIM (and that is not the goal here anyways). Nevertheless, we intend to investigate whether discarding small changes in pressure after a few iterations of the nonlinear loop, will affect convergence of the FIM.

We test the proposed strategy on the numerical example outlined in section 3.1. The pressure is only updated in the first three Newton iterations. Figure 3.2 shows variation of L_∞ norm of the transport residual with respect to iteration number. Based on the figure it can be inferred that the nonlinear solver fails to converge due to the stagnant residual that does not seem to change as iterations proceed. This numerical experiment indicates that although pressure iterates tend to change

slightly after the first few iterations, but updating consistently throughout the course of Newton iteration is necessary for convergence. Saturation and pressure changes over Newton iterations highly depend on each other due to the tight coupling of flow and transport equations. The dependence of saturation on pressure is to the extent that freezing the pressure field at any point during Newton flow causes saturation iterates to become stagnant, resulting in a fixed residual thereafter.

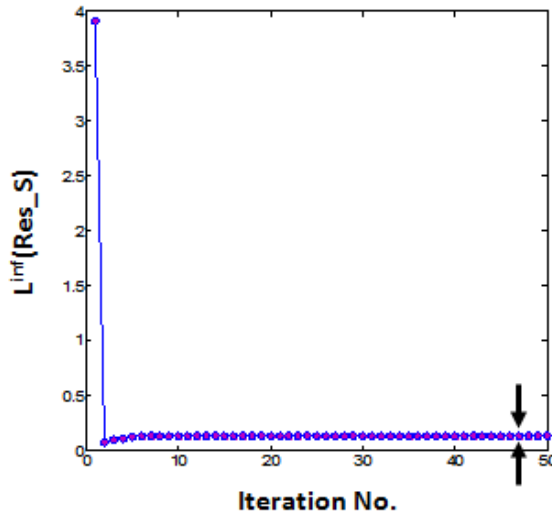


Figure 3.2: Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.

3.2.2 Updating pressure only where total velocity changes exceed a specified threshold

Our second updating strategy is based on the idea of updating pressure only in areas where changes in the total volumetric flux (total velocity) exceed a specified threshold over Newton iterations. The total velocity is obtained from summing up individual

phase volumetric fluxes. In order to test this strategy, we propose the following algorithm. At each Newton iteration, after the linear system is solved and the variables (pressure and saturation) are updated, compute total velocities at cell interfaces. Calculate the relative change of total velocities over the Newton iteration. Loop through all the cells. For each cell, consider the corresponding interfaces and check if the relative change of total velocities over the current Newton iteration is greater than the specified threshold, ϵ . If so, save the pressure solution of the cell, otherwise discard the pressure iterate and stick to the old iterate (see Algorithm 3). We define the relative change of total velocity over Newton iteration ν for a given cell i as

$$\delta^* u_{t,i}^{\nu+1} = \max \left\{ \left| \frac{u_{t,x}^{\nu+1} - u_{t,x}^{\nu}}{u_{t,x}^{\nu}} \right|_i, \left| \frac{u_{t,y}^{\nu+1} - u_{t,y}^{\nu}}{u_{t,y}^{\nu}} \right|_i, \left| \frac{u_{t,z}^{\nu+1} - u_{t,z}^{\nu}}{u_{t,z}^{\nu}} \right|_i \right\}, \quad (3.2)$$

where $u_{t,x}$, $u_{t,y}$ and $u_{t,z}$ denote total velocities in x , y and z directions respectively. In the presence of gravity, the phase flow directions might change over Newton iterations. This phenomenon is referred to as ‘phase flow reversal’ or phase ‘flip flop’. Once phase flip flop occurs, the phase upstream direction at the corresponding interface changes. Phase flip flop leads to discontinuity in the numerical flux function. To better understand the impacts of phase flip flop on the numerical flux functions, consider the following scenario. At some iteration ν , the phase flow directions over the interface between two adjacent cells, i and $i+1$, are shown in Figure 3.3. At this iteration, given the phase flow directions, the numerical flux function for water phase is defined as (Brenier and Jaffre (1991))

$$F_{w,i+1/2} = \frac{\lambda_w(S_i)}{\lambda_w(S_i) + \lambda_o(S_{i+1})} (q_T - K_{i+1/2} \lambda_o(S_{i+1}) \Delta \rho g). \quad (3.3)$$

Suppose, at the next iteration, the pressure of cell i changes slightly, such that cell i is no longer the upstream for water phase. With the new upstream directions, the

numerical flux for water at the interface is defined as (Brenier and Jaffre (1991))

$$F_{w,i+1/2} = \frac{\lambda_w(S_{i+1})}{\lambda_w(S_{i+1}) + \lambda_o(S_{i+1})} (q_T - K_{i+1/2} \lambda_o(S_{i+1}) \Delta \rho g). \quad (3.4)$$

Comparison of Equations 3.3 and 3.4 shows that slight changes in the pressure could lead to discontinuity of the numerical flux over the interface. This is due to the fact that the mobility terms corresponding to the phase with flow reversal, are evaluated at different grid cells at iterations ν and $\nu + 1$. Discontinuities in the flux function could lead to significant changes in the phase velocities (hence the total velocity), and cause oscillations in the residual.

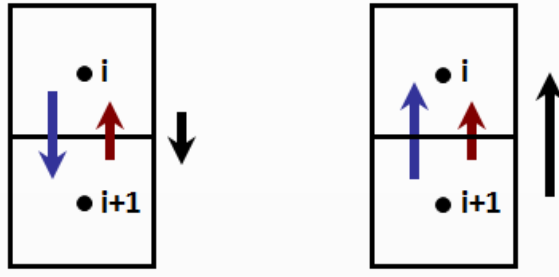


Figure 3.3: The figures on the left and right show the phase flow directions at the beginning and at the end of a hypothetical FIM iteration respectively. The phase flow reversal over Newton iterations is sometimes referred to as phase flip flop. This phenomenon results in discontinuity of flux function over iterations due to change in the upstream directions and could lead to oscillations in residual and convergence failure.

Algorithm 3 Algorithm for adaptive pressure update

```

1: while convergence criterion is not met do
2:   Compute the full Jacobian  $\mathbf{J}^\nu$  and residual  $\mathbf{r}^\nu$  ;
3:   Solve for Newton update from  $\mathbf{J}^\nu \delta \mathbf{x}^\nu = -\mathbf{r}^\nu$  ;
4:   Compute  $\mathbf{x}^{\nu+1} = \mathbf{x}^\nu + \delta \mathbf{x}^\nu$  ;
5:   Compute  $\mathbf{u}_T^{\nu+1}$  ;
6:   for  $i = 1 \rightarrow N_{cells}$  do
7:     if  $\max_{i=1, \dots, N_{cells}} \{\delta^* u_{t,i}\} \geq \epsilon$  then
8:       Save  $p_i^{\nu+1}$  ;
9:     else
10:       $p_i^{\nu+1} := p_i^\nu$  ;
11:    end if
12:  end for
13:   $\nu := \nu + 1$  ;
14: end while

```

We test the proposed pressure updating strategy on an example similar to the one outlined in 2.7.2. The reservoir is discretized using $10 \times 1 \times 10$ cells in x , y and z directions respectively. Figure 3.5 shows the cells with phase flip flop over the six sample Newton iterations. In the early iterations, phase flip flop occurs more frequently but as we approach the solution, it occurs in fewer cells. Phase flow directions eventually become steady at later iterations close to convergence.

Figure 3.4 on the left shows plot of number of cells with phase flip flop with respect to the iteration number. The figure on the right presents plot of number of cells with pressure updates with respect to the iteration number. Comparison of these two figures shows the relationship between phase flip flop and variation of total velocities. In early Newton iterations, phase flip flops affect a big chunk of the domain (up to almost 80% of the entire cells at iteration 13 for instance). Frequent change

of upstream directions due to phase flip flops, results in changes in total velocities that exceed the specified threshold, $\epsilon = 0.0001$ in all of the cells. Thus, based on the proposed algorithm we need to update pressure in the entire domain. As soon as the number of cells with flip flops begins to decline at later iterations (shown by the arrow in the figure), fewer pressure updates are required.

Figures 3.5 and 3.6 portray the maps of cells with phase flip flop and cells that require pressure updates at six sample Newton iterations respectively. Comparison of these figures shows that at a given iteration, all of the cells with phase flip flop require pressure updates since the relative change in total velocities in their corresponding interfaces is greater than the specified value.

The change of L_∞ norm of the transport residual with respect to the number of Newton iteration is presented in Figure 3.7. The oscillations in the residual due to phase flip flop are apparent in this figure. As soon as phase flow directions begin to stabilize (marked by the arrow), the residual starts to decline and eventually it becomes stagnant. The stabilized L_∞ norm of the residual, which we refer to it as “the final error”, happens to meet the convergence criterion defined here and therefore convergence is obtained. Notice that in this example we chose a tight criterion for selective pressure update by setting $\epsilon = 0.0001$. Increasing parameter ϵ gives rise to greater final errors. The dependence of the final error on ϵ for this particular example is shown in Figure 3.8. By picking a greater ϵ , we decrease the number of pressure updates in each iteration, since by definition a greater change in total velocity is required to update pressure.

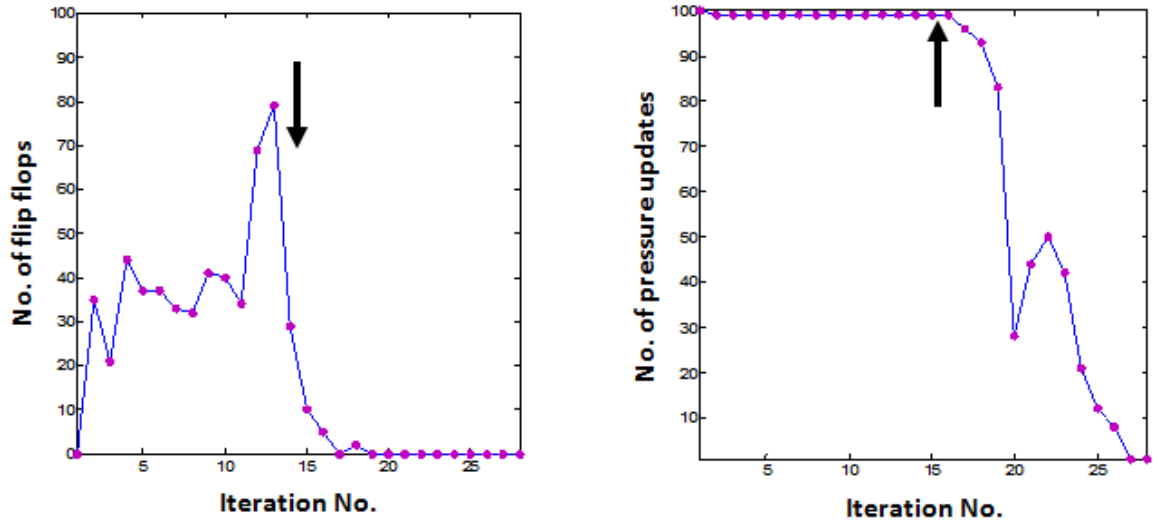


Figure 3.4: Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.

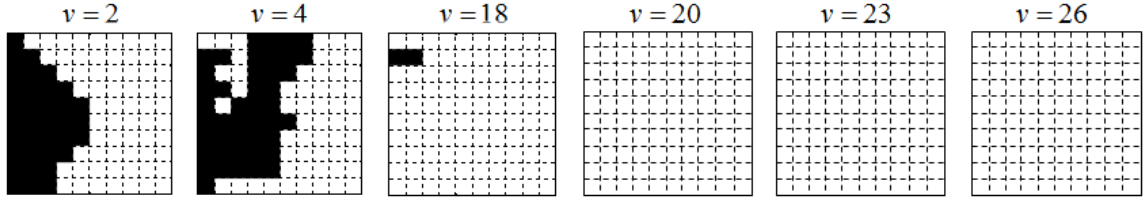


Figure 3.5: The map of cells that have phase flip flop at the corresponding interfaces for six sample Newton iterations, $\nu = 2, 4, 18, 20, 23$ and 26 . In early Newton iterations, phase flip flops occur more frequently but phase flow directions tend to become steady as we approach solution in later iterations.

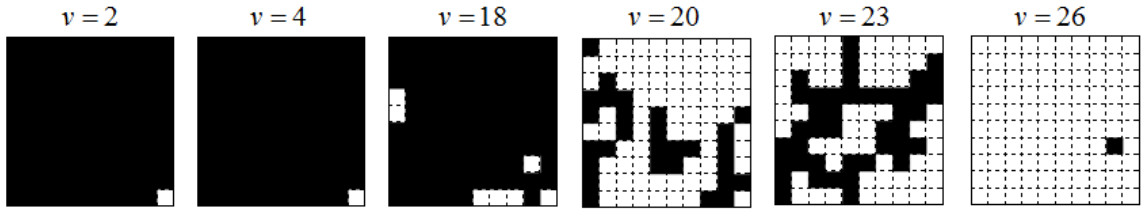


Figure 3.6: The map of cells that have variations of total velocities at the corresponding interfaces greater than the specified threshold, $\epsilon = 0.001$ for six sample Newton iterations, $\nu = 2, 4, 18, 20, 23$ and 26 . These are the cells that require pressure updates. As we proceed with the nonlinear iterations, fewer pressure updates are needed.

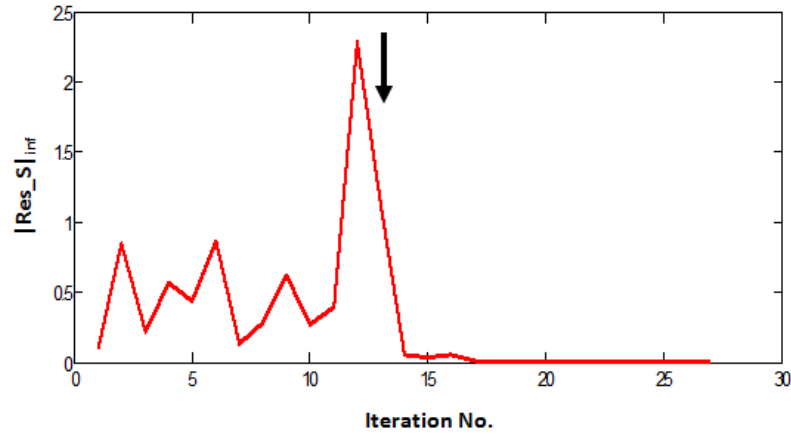


Figure 3.7: L_∞ norm of the residual of saturation equations with respect to the iteration number. Initially the residual has oscillations due to phase flip flops. As soon as flow directions become steady (marked by the arrow), the residual begins to decline and eventually it becomes stagnant. The final residual norm in this example with $\epsilon = 0.001$, happens to meet the convergence criterion and therefore convergence is obtained.

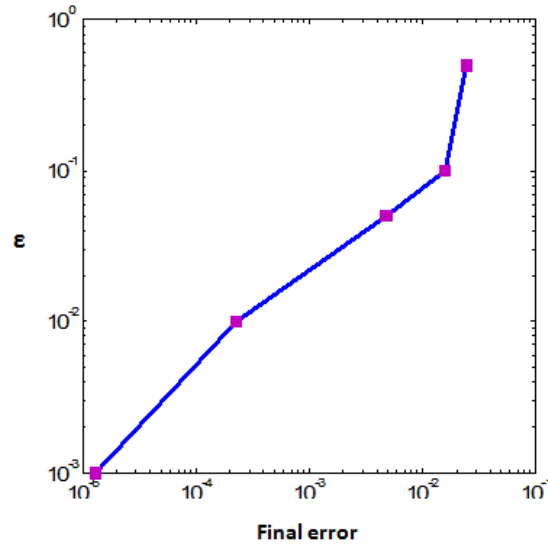


Figure 3.8: Plot of parameter ϵ versus the stabilized L_∞ norm of the residual, referred to as final error. By specifying a greater ϵ , the number of pressure updates in each iteration decreases, since a greater change in total velocity is required to update pressure. Less frequent pressure updates give rise to stagnant residuals with larger L_∞ norms.

3.3 Phase-based ordering

Phase-based ordering refers to the process of performing a reordering of conservation equations based on phase potentials. In order to understand the advantage of using phase-based ordering, consider the nonlinear system of equations arising from finite volume discretization of governing equations for the two-phase flow system outlined in section 1.1:

$$\begin{aligned} f_{w1}(S_{w1}, \dots, S_{wN}, p_1, \dots, p_N) &= 0 \\ f_{w2}(S_{w1}, \dots, S_{wN}, p_1, \dots, p_N) &= 0 \\ &\vdots \\ f_{wN}(S_{w1}, \dots, S_{wN}, p_1, \dots, p_N) &= 0. \end{aligned} \tag{3.5}$$

In the above equations, it is apparent that although we use phase-based upstreaming for discretization, the dependence of conservation equations on saturation variables does not exhibit a special pattern. Based on phase-based upstreaming, the numerical flux at a given interface for each phase depends only on the saturation of the upstream cell. The upstream direction of a phase at a given interface is determined by comparing the phase potentials of the two cells sharing the interface. Potential of phase p at any grid cell k is defined as

$$\Phi_{p,k} = p_k - \gamma_p z_k, \tag{3.6}$$

where γ_p and z_k are the specific gravity of phase p and depth of cell k respectively. It is possible to provide a more specific pattern for saturation unknowns in conservation equations by reordering equations 3.5 based on phase potentials. Suppose we define a new permutation of cells, $\sigma_1, \dots, \sigma_N$ such that $\Phi_{w,\sigma,i} \geq \Phi_{w,\sigma,j}$ whenever $i < j$. We use the σ ordering to order all the water conservation equations and the associated

variables S_w . The resulting nonlinear system in this case looks like

$$\begin{aligned}
 f_{w,\sigma_1}(S_{w,\sigma_1}, p_1, \dots, p_N) &= 0 \\
 f_{w,\sigma_2}(S_{w,\sigma_1}, S_{w,\sigma_2}, p_1, \dots, p_N) &= 0 \\
 &\vdots \\
 f_{w,\sigma_N}(S_{w,\sigma_1}, \dots, S_{w,\sigma_N}, p_1, \dots, p_N) &= 0.
 \end{aligned} \tag{3.7}$$

As Equation 3.7 shows, reordering of the nonlinear system based on Φ_w leads to a triangular pattern for saturation variables in the system of equations. This is because the flux terms depend only on saturation of upstream cells. Thus, when we march down from upstream (the cell with the highest potential) to downstream (the cell with the lowest potential), the triangular pattern is obtained. This property allows us to perform a forward substitution and solve a series of scalar nonlinear conservation equations for saturation variables, given that pressure solution $p_1, \dots, p_N$ is available. The triangular pattern in the equations manifests itself in the Jacobian matrix as well. The linearized set of water and oil conservation equations has the form

$$\begin{bmatrix} \mathbf{J}_{\mathbf{ww}} & \mathbf{J}_{\mathbf{wp}} \\ \mathbf{J}_{\mathbf{ow}} & \mathbf{J}_{\mathbf{op}} \end{bmatrix} \begin{bmatrix} \delta \mathbf{S}_w \\ \delta \mathbf{p} \end{bmatrix} = - \begin{bmatrix} \mathbf{r}_w \\ \mathbf{r}_o \end{bmatrix}, \tag{3.8}$$

where $\mathbf{r}_w$ and $\mathbf{r}_o$ are residuals of water and oil conservation equations, and the Jacobian matrix $\mathbf{J}$ is made up of four blocks, $\mathbf{J}_{\mathbf{ww}}$, $\mathbf{J}_{\mathbf{wp}}$, $\mathbf{J}_{\mathbf{ow}}$ and $\mathbf{J}_{\mathbf{op}}$ with the same size $N \times N$. These blocks represent derivatives of water and oil residuals with respect to pressure and saturation variables. Due to reordering of equations, $\mathbf{J}_{\mathbf{ww}}$ is lower triangular. Notice that $\mathbf{J}_{\mathbf{ow}}$ does not necessarily have a triangular form since we ordered the equations based on potential of water phase and the upstream directions for oil and water phases are not the same in all interfaces due to counter-current flow. However, it is possible to introduce a reordering based on potential of oil phase for oil

conservation equations (resulting in a cell permutation different from σ), such that $\mathbf{J}_{\mathbf{ow}}$ becomes lower triangular. In two-phase flow systems, applying only one ordering based on potential of one of the phases is enough for solving the equations.

3.4 Reduced Newton method

It is possible to reformulate conservation equations into a reduced system of smaller size that depends only on pressure variables. The reduced system is obtained from the Schur complement reduction of the original system 3.8 as follows. We first express saturation variables as a function of pressure variables using water conservation equations given in the upper half of system 3.8, as follows

$$\delta \mathbf{S}_w = -\mathbf{J}_{ww}^{-1} (\mathbf{J}_{wp} \delta \mathbf{p} + \mathbf{r}_w). \quad (3.9)$$

We then plug equation 3.9 into oil conservation equations (the lower half of system 3.8). The resulting reduced system looks like

$$(\mathbf{J}_{op} - \mathbf{J}_{ow} \mathbf{J}_{ww}^{-1} \mathbf{J}_{wp}) \delta \mathbf{p} = -(\mathbf{r}_o - \mathbf{J}_{ow} \mathbf{r}_w). \quad (3.10)$$

As Equation 3.10 shows, the reduced system has a Jacobian with a smaller size which is much cheaper to solve as compared to the full system. The reduced Jacobian has the form

$$\mathbf{J}_{reduced} = \mathbf{J}_{op} - \mathbf{J}_{ow} \mathbf{J}_{ww}^{-1} \mathbf{J}_{wp}. \quad (3.11)$$

Algorithm 4 shows the reduced Newton algorithm for solving two-phase flow and transport equations. At each Newton iteration, the Schur complement reduction is employed to update pressure by solving the reduced linear system 3.10. Once pressure is solved for, phase-based ordering is applied and saturation variables are updated

cell-by-cell, from upstream to downstream, by solving nonlinear scalar transport equations. The nonlinear loop is repeated until convergence is obtained. As compared to the fully coupled FIM, where pressure and saturation variables are updated simultaneously, reduced Newton method is significantly superior in terms of computational efficiency if convergence is obtained.

Algorithm 4 Algorithm for solving two-phase flow and transport equations using reduced Newton method.

- 1: **while** convergence criterion is not met **do**
 - 2: Compute the full Jacobian, $\mathbf{J}(S_w^\nu, p^\nu)$;
 - 3: Solve $(\mathbf{J}_{\text{op}} - \mathbf{J}_{\text{ow}}\mathbf{J}_{\text{ww}}^{-1}\mathbf{J}_{\text{wp}})^\nu \delta \mathbf{p}^\nu = -(\mathbf{r}_{\text{o}} - \mathbf{J}_{\text{ow}}\mathbf{r}_{\text{w}})^\nu$;
 - 4: Compute $\mathbf{p}^{\nu+1} = \mathbf{p}^\nu + \delta \mathbf{p}^\nu$;
 - 5: Compute $\Phi_{\text{w}}^{\nu+1} = \mathbf{p}^{\nu+1} - \gamma_{\text{w}}\mathbf{z}$;
 - 6: Reorder equations based on $\Phi_{\text{w}}^{\nu+1}$;
 - 7: Compute $\mathbf{S}_{\text{w}}^{\nu+1}$ by solving nonlinear transport equations cell-by-cell, from upstream to downstream ;
 - 8: $\nu := \nu + 1$;
 - 9: **end while**
-

3.4.1 Convergence behavior of reduced Newton method

Rigorous mathematical analysis proves that reduced Newton method converges for any timestep size in the case of co-current flow problems (Kwok and Tchelepi (2007)). However, when gravity dominates and counter-current flow occurs, the algorithm is no longer unconditionally convergent. In order to present the impact of gravity on convergence behavior of reduced Newton method, we set up the following example. Consider a quarter 5-spot homogeneous model in the $x-z$ cross section where gravity is acting in the z direction. Initially, the top half of the domain is filled with water and the lower half is filled with oil. Injection and production conditions are similar to the

ones used in examples 2 to 6 in the previous chapter. We compare two scenarios, a) there is no density difference between water and oil phases, thus there are no counter-current flow effects, and b) there is a significant difference in the density of water and oil phases. We use the reduced Newton method to solve the governing equations in both scenarios. Figure 3.9 shows plots of number of Newton iterations required for convergence versus timestep size for both problems. The blue plot, corresponding to no gravity (pure co-current flow) case, shows that reduced Newton method converges for all of the timesteps used in this example ($\Delta t = 1, 10, 100, 1000$ cell PVI). This is in agreement with theory that proves unconditional convergence for the co-current flow case. The red plot corresponds to the second scenario where significant counter-current flow exists due to gravity effects. For this case, the reduced Newton method does not converge for ($\Delta t \geq 10$ cell PVI). Figures 3.10 and 3.11 present saturation maps for the two scenarios obtained using $\Delta t = 10$ cell PVI for the gravity-dominated case and $\Delta t = 1000$ cell PVI for the pure co-current flow case. The saturation map of the second case clearly shows counter-current flow due to fluids density difference. In order to pinpoint the problem of poor convergence of reduced Newton method in the presence of counter-current flow, we need to focus on the coupling between flow and transport equations. Reduced Newton method allows us to update pressure and saturation equations in a decoupled fashion. In other words, once we update pressure variables from the reduced linear system, we freeze the pressure solution and use it to calculate the phase potentials based on which we reorder conservation equations. Once transport equations are reordered, we update saturations on a cell-by-cell basis. In the co-current flow case, the phase upstream directions that are used to update pressure variables are essentially consistent with the phase-based ordering that is used to update saturation variables. This is due to the fact that in a co-current setting, all phases flow in the same direction which does not change as Newton iterations proceed. In this case, phase upstream directions before and

after pressure update are the same. Just on the contrary, in counter-current flow case, phase flow directions change over Newton iterations. This means that at some cell interfaces, phase flow directions change once pressure is updated and therefore the ordering based on which we update saturation variables is not consistent with the phase upstream directions based on which we update pressure variables. The mismatch between flow and transport equations causes convergence failure. Notice that as the coupling between flow and transport complicates for larger timesteps, the phase flip flop at cell interfaces happen more frequently and therefore convergence is more likely to fail.

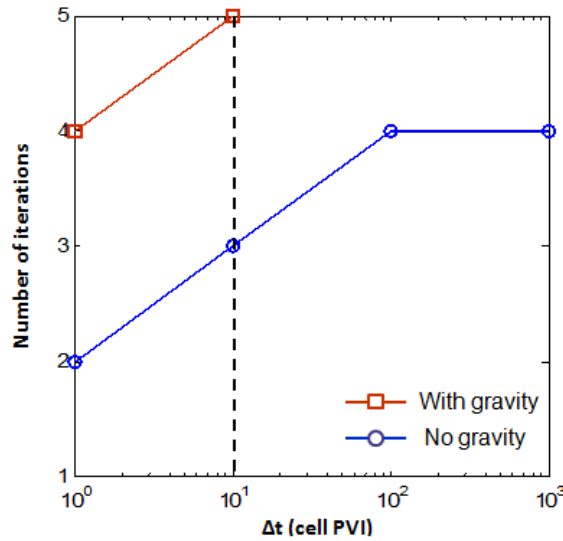


Figure 3.9: Convergence behavior of reduced Newton method for two problems: a) without gravity and effects, thus co-current flow everywhere (the blue plot), for which the algorithm converges unconditionally, and b) with gravity and counter-current flow effects (the red plot), for which the algorithm converges for $\Delta t \leq 10$ cell PVI.

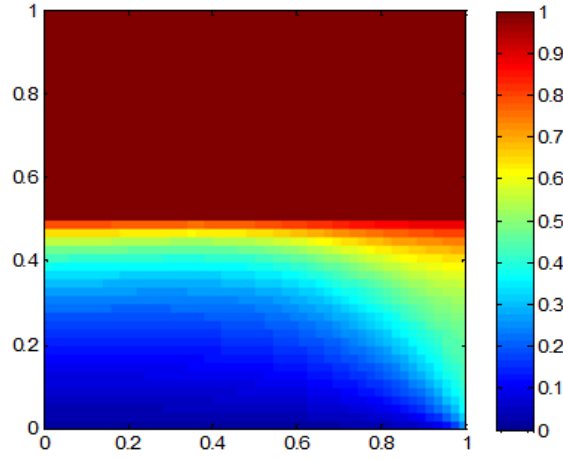


Figure 3.10: Saturation map at $t = \Delta t = 1000$ cell PVI, corresponding to the displacement problem with no gravity effects (pure co-current flow), obtained using reduced Newton method. For this extremely large timestep, convergence is obtained in 4 iterations.

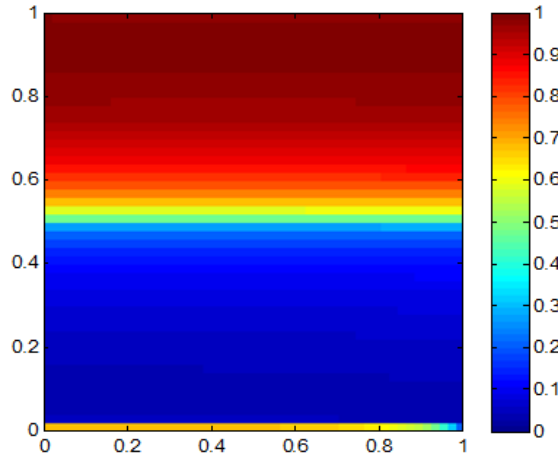


Figure 3.11: Saturation map at $t = \Delta t = 10$ cell PVI, corresponding to the displacement problem with gravity effects due to density difference of fluids, obtained using reduced Newton method. The timestep used here is the largest for which convergence is obtained.

Chapter 4

Efficient Coupling of Flow and Transport

The main purpose of this chapter is to provide a solution strategy to improve the robustness and efficiency of the fully coupled FIM. It is well known that the cost associated with updating all the unknowns simultaneously, everywhere and all the time, can be quite expensive. Conventional sequential-implicit strategies can be used to reduce the cost, but they suffer from severe restrictions on the allowable timestep size. Based on the new strategy proposed in this chapter, we no longer need to update the unknowns in a fully coupled fashion, everywhere and all the time. The crux of the algorithm comes in using a hybrid strategy to ensure that coupling is resolved in an adaptive yet efficient fashion by treating co-current and counter-current flow regions differently. Based on the new solution strategy, we update pressure and saturation simultaneously only in regions where counter-current flow is present. We identify these regions by constructing the so-called phase flow graph and decomposing it into its strongly connected components at each Newton iteration.

This chapter is organized as follows. First, we provide an overview of the basic concepts in graph theory. In particular, we review the notion of strongly connected

components and the corresponding algorithms. We then introduce the concept of phase flow graphs and demonstrate how graph theory can be used to identify the counter-current flow regions. In the rest of the chapter, we describe the new solution strategy, i.e. the adaptively coupled hybrid method (ACHM), for solving coupled flow and transport equations. We compare the new nonlinear solver with the current state-of-the-art for a wide range of problems with special focus on counter-current flow due to buoyancy.

4.1 Graph theory

In mathematics and computer science graphs are used to model pairwise relations between objects from a certain collection. The graph $G = (V, E)$ in this context represents a collection of nodes (vertices) V and a collection of edges E that connect pairs of nodes. In the following we review some of the basic concepts in graph theory that we will use throughout the chapter.

A graph may be *undirected* or *directed*. In an undirected graph, there is no distinction between the nodes associated with each edge, whereas in a directed graph the edges are directed from one node to another (see Figure 4.1).

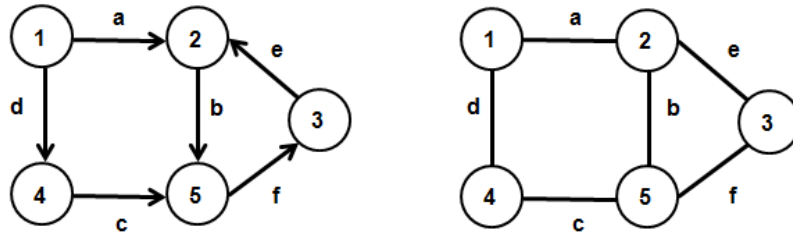


Figure 4.1: The figure shows a directed graph on the left and an undirected graph on the right. Both graphs have the same number of nodes and edges, but have entirely different structures.

A given directed or undirected graph can be represented as a collection of adjacency lists or as an adjacency matrix. The *adjacency list representation* of a graph $G = (V, E)$ consists of an array of $|V|$ lists, one for each node in V . For each $u \in V$, the adjacency list $Adj[u]$ contains all the vertices v such that there is an edge $(u, v) \in E$. That is, $Adj[u]$ consists of all the vertices adjacent to u in G . Adjacency list representation is usually the method of choice for sparse graphs, those for which $|E| \ll |V|^2$. For the *adjacency matrix representation* of a graph, we assume that the vertices are numbered $1, 2, \dots, |V|$ in some arbitrary manner. Then the adjacency matrix representation of a graph G consists of a $|V| \times |V|$ matrix $A = (a_{ij})$ such that

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (4.1)$$

The adjacency matrix representation is used mostly when the graph is dense. Figure 4.2 shows the adjacency list and the adjacency matrix representations of the directed graph presented in Figure 4.1. The preferred way of representing this graph is the adjacency list due to the sparsity of the graph.

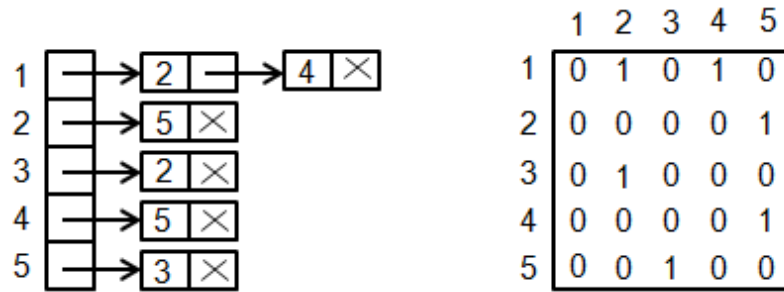


Figure 4.2: The adjacency list (on the left) and the adjacency matrix (on the right) representations of the directed graph presented in Figure 4.1.

A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. A finite path always has a first vertex,

called its start vertex, and a last vertex, called its end vertex. These two vertices are sometimes referred to as terminal vertices. The other vertices in the path are internal vertices. A *cycle* is a path such that the start vertex and end vertex are the same. The choice of the start vertex in a cycle is arbitrary. A *subgraph* of a graph G is a graph whose vertex set is a subset of that of G , and whose adjacency relation is a subset of that of G restricted to this subset.

An undirected graph is said to be connected if there is a path from any vertex to any other vertex in the graph. A graph that is not connected is said to be disconnected. A directed graph is called *weakly connected* if replacing all of its directed edges with undirected edges produces a connected (undirected) graph. It is *strongly connected* or *strong* if it contains a directed path from u to v and a directed path from v to u for every pair of vertices u, v .

The *connected components* of an undirected graph G are the sets of vertices such that any two vertices in each set are connected, and no two vertices in different sets are connected. The ***strongly connected components*** of a directed graph G are its maximal strongly connected subgraphs. If each strongly connected component is condensed to a single vertex, the resulting graph is a *directed acyclic graph*. Notice that every strongly connected component contains at least one directed cycle. Thus a directed graph is acyclic, if and only if it has no strongly connected components with more than one vertex.

4.2 Phase flow graph

By definition, *phase flow graph* $G = (V, E)$ refers to the directed graph constructed using grid cells as vertices and individual phase velocities at cell interfaces as edges. Notice that in the case of multiphase flow, no distinction is made between velocities of different phases as far as constructing the graph edges is concerned. For instance,

in a water-oil system, there are two directed edges between any two adjacent vertices representing velocity vectors of water and oil phases at the corresponding shared interface between the two grid cells. At each Newton iteration, an adjacency list representation of the phase flow graph is readily obtained at no extra cost using the cells connections list and phase potentials in grid cells.

Phase flow graphs contain information about phase flow directions. Suppose graph G represents the phase flow graph at some instant in time or Newton iteration. By decomposing G into the strongly connected components, we can identify the counter-current flow regions in a systematic yet efficient way. The shape and structure of strongly connected components of G is directly related to phases flow directions at cell interfaces. The structure of strongly connected components of phase flow graph is directly related to phases flow directions at cell interfaces. In the following, we consider three possible scenarios for a two-phase system. In each case, we investigate the structure of strongly connected components of the corresponding phase flow graph.

4.2.1 Pure co-current flow

Pure co-current flow refers to the case where both phases flow in the same direction at all cell interfaces. In the case of pure co-current flow, each cell by itself constitutes a strongly connected component. We refer to such components as *single-cell components* hereafter. Figure 4.3 presents an example of pure co-current flow and the corresponding single-cell components.

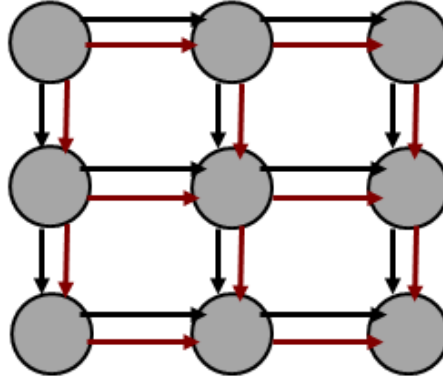


Figure 4.3: In the case of pure co-current flow, each cell is a strongly connected component of phase flow graph (shown in grey).

4.2.2 Pure counter-current flow

Pure counter-current flow refers to the case where phases flow in opposite directions at all cell interfaces. In this case, regardless of the number of cells, the phase flow graph contains only one strongly connected component which covers the entire domain (see Figure 4.4).

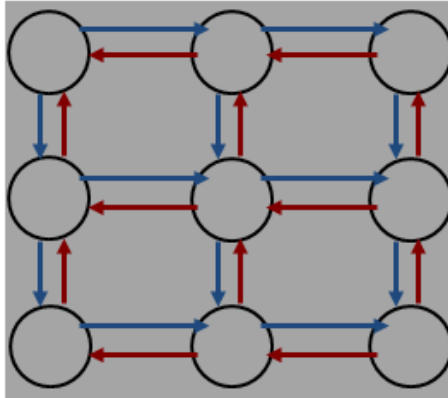


Figure 4.4: In the case of pure counter-current flow, only one strongly connected component exists (shown in grey), which is made up of the entire grid cells.

4.2.3 Mixed flow

Mixed flow refers to the case where phases flow in the same direction in some cell interfaces and in opposite directions in other cell interfaces. In this case, strongly connected components might vary from *single-cell* components to large *multi-cell* components (see Figure 4.5). Single-cell components denote co-current flow regions whereas multi-cell components characterize counter-current flow regions.

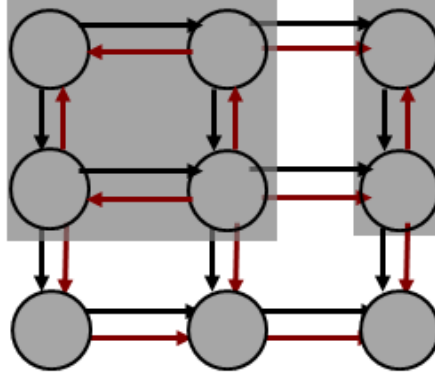


Figure 4.5: In the case of mixed flow where co-current and counter-current flow interfaces coexist, strongly connected components vary from single-cell to multi-cell components. The phase flow graph shown in this example is decomposed into five strongly connected components, two of which are made up of multiple cells and the rest are single-cell components indicating counter-current and co-current flow regions respectively.

There are several algorithms available in the literature for decomposing a directed graph into strongly connected components. Most of these algorithms directly apply the graph traversal methods such as depth-first search (DFS) and breadth-first search (Cormen et al. (2009)). In this study, we use *Tarjan's algorithm* because

- It is of the order $O(|V| + |E|)$ and thus it runs in linear-time.
- It requires only one DFS.

- The algorithm provides a reverse topological ordering of the strongly connected components at no extra cost.

These properties make Tarjan's algorithm a good candidate for decomposing the graphs into strongly connected components and traversing the resulting components very efficiently.

4.3 Adaptively coupled hybrid method (ACHM)

In order to provide an efficient yet robust nonlinear solver for coupled flow and transport equations, we need to address the following issues that impact the efficiency or robustness of the state-of-the-art solution strategies,

- **Computational cost:** Simultaneous update of variables, everywhere and all the time, is computationally expensive and often times unnecessary. This is the main drawback of the standard Newton's method.
- **Adaptivity:** An efficient alternative to the fully coupled FIM should resolve the coupling between flow and transport equations in an adaptive fashion during the course of Newton iterations.
- **Inherent nonlinearities:** Convergence behavior of the nonlinear solver is key to robustness. Addressing the inherent nonlinearities properly gives rise to better convergence behavior.

To address all of the aforementioned difficulties, we propose the ACHM, as an alternative to the standard Newton's method. Algorithm 5 presents the pseudocode of the proposed nonlinear solver. At each Newton iteration, the algorithm is composed of three consecutive steps including pressure update, identifying strongly connected

regions, and sequential saturation update and safeguard (see Figure 4.6). In the following, we describe these steps in detail (Shahvali and Tchelepi (2013)).

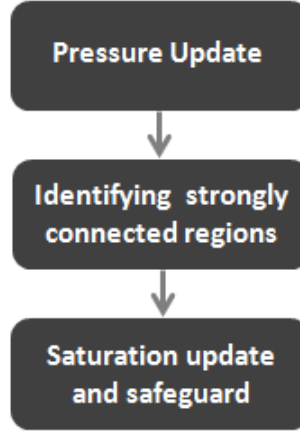


Figure 4.6: At each Newton iteration, the adaptively coupled hybrid method (ACHM) consists of three consecutive steps.

4.3.1 Pressure update

At each Newton iteration, we update pressure by solving the reduced linear system obtained from Schur complement reduction of the full system of equations as outlined in section 2.4. Thus the first step of ACHM is identical to pressure updating strategy used in reduced Newton algorithm (Kwok and Tchelepi (2007)). As noted earlier, the computational cost associated with solving the reduced system for pressure is much less than that of solving the full system for updating pressure and saturation simultaneously.

4.3.2 Identifying strongly connected regions

The second step of the hybrid algorithm is to identify co-current and counter-current flow regions. The reason why is because we intend to apply different updating strategies for these regions. We accomplish this task by constructing the phase flow graph

and decomposing it into strongly connected components using Tarjan’s algorithm. As noted earlier, counter-current flow regions are characterized by multi-cell connected components. By condensing the cells corresponding to each multi-cell component into a ‘supernode’, we form the corresponding directed acyclic graph. The resulting directed acyclic graph can be readily traversed from upstream to downstream at no extra cost since Tarjan’s algorithm provides a reverse topological ordering of the strongly connected components.

4.3.3 Sequential saturation update and safeguard

In order to update saturation variables we proceed as follows. We first create a *directed acyclic graph* by condensing the cells corresponding to each multi-cell component into a ‘supernode’. We traverse the directed acyclic graph from upstream to downstream in a sequential fashion. Notice that decomposing the phase flow graph into strongly connected components and topological sorting of the corresponding directed acyclic graph are done concurrently using Tarjan’s algorithm. As we visit components, we treat them differently depending on whether they are made up of a single cell or multiple cells. For a single-cell component, we accept the pressure solution obtained in the pressure update step and we update saturation nonlinearly by solving the nonlinear scalar transport equation for that cell.

For a multi-cell component, we first discard the pressure iterates obtained in the pressure update step. We then compute the local Jacobian and local residual, $\mathbf{J}_c$ and $\mathbf{r}_c$ corresponding to the component’s grid cells, and update pressure and saturation variables of these cells simultaneously by solving the local linear system $\mathbf{J}_c \delta \mathbf{x}_c = \mathbf{r}_c$.

Notice that by applying fully coupled (simultaneous) updating strategy in the multi-cell connected regions, we ensure that the existing tight coupling of flow and transport in these regions is addressed properly. On the other hand, since saturation variables are obtained from scalar transport equations in co-current flow regions, the

computational cost might reduce significantly as compared to the standard Newton, where pressure and saturation are updated simultaneously everywhere. The shape and size of the strongly connected components might change significantly over the course of Newton iterations. At each Newton iteration, the algorithm addresses the coupling of variables on these components in an adaptive fashion. The updating strategies range from one extreme being global single-cell saturation updates (reduced Newton method) to the other extreme of global simultaneous pressure and saturation update (fully coupled FIM).

The last component of the ACHM framework is the safeguard strategy for saturation iterates. As we solve for saturation of single-cell and multi-cell components sequentially, we safeguard the iterates using the method based on trust regions of flux functions, as outlined in section 2.3.

Algorithm 5 Adaptively coupled hybrid framework (ACHM), for solving two-phase flow and transport equations

- 1: **while** convergence criterion is not met **do**
 - 2: Compute the full Jacobian, $\mathbf{J}(S_w^\nu, p^\nu)$;
 - 3: Solve the reduced linear system $(\mathbf{J}_{\text{op}} - \mathbf{J}_{\text{ow}}\mathbf{J}_{\text{ww}}^{-1}\mathbf{J}_{\text{wp}})^\nu \delta \mathbf{p}^\nu = -(\mathbf{r}_\text{o} - \mathbf{J}_{\text{ow}}\mathbf{r}_\text{w})^\nu$ for pressure ;
 - 4: Compute $\mathbf{p}^{\nu+1} = \mathbf{p}^\nu + \delta \mathbf{p}^\nu$;
 - 5: Decompose the phase flow graph into strongly connected components and construct the corresponding directed acyclic graph G ;
 - 6: **for** $c = \text{upstream of } G \rightarrow \text{downstream of } G$ **do**
 - 7: **if** c is a single-cell component **then**
 - 8: Update saturation by solving the nonlinear scalar transport equation ;
 - 9: Safeguard saturation iterate ;
 - 10: **else**
 - 11: Discard pressure iterates corresponding to the component's cells ;
 - 12: Compute the local Jacobian $\mathbf{J}_\mathbf{c}$ and the local residual $\mathbf{r}_\mathbf{c}$;
 - 13: Update pressure and saturation of the cells by solving $\mathbf{J}_\mathbf{c}\delta \mathbf{x}_\mathbf{c} = \mathbf{r}_\mathbf{c}$;
 - 14: Safeguard saturation iterates ;
 - 15: **end if**
 - 16: **end for**
 - 17: $\nu := \nu + 1$;
 - 18: **end while**
-

4.4 Numerical examples

To test the efficiency of the proposed nonlinear solver, we use a variety of numerical examples in 2D and 3D. The results for the 2D examples are obtained with a MATLAB code that we implemented for testing the algorithm. For the 3D example, we implemented the algorithm inside the Automatic Differentiation-General Purpose Research Simulator (AD-GPRS) that was developed in Stanford University's SUPRI-B research group by Cao (2002), Younis (2011) and Zhou (2012). AD-GPRS is used for in-house research by SUPRI-B group members as well as other research groups and companies. In each example, we pay special attention to counter-current flow due to buoyancy and compare the efficiency of the new solver with the current state-of-the-art.

4.4.1 Homogeneous 2D reservoir with gravity - uniform initial saturation condition

In the first example, we consider a homogeneous 2D quarter 5-spot domain. The reservoir is discretized using $50 \times 1 \times 50$ cells in x , y and z directions respectively. Water is injected from the top left corner at constant rate $q = 1$, and the producer is located at the bottom right corner producing at constant pressure $p = 4000$. The initial water saturation is $S_w = 0.001$. Water/oil viscosity ratio and density ratio are 0.3 and 2 respectively.

Figure 4.7 shows the saturation map at the end of simulation, i.e. $t = 100$ cell PVI. The simulation was run for only one aggressive timestep, $\Delta t = 100$ cell PVI, using ACHM. The saturation map shows that the majority of the reservoir is not touched by the injected water due to the high density contrast between the fluids, causing the injected water to travel towards the bottom of the reservoir. For this problem updating pressure and saturation variables simultaneously using the standard

Newton's method seems to be unnecessary.

Figure 4.8 shows maps of strongly connected components at different iterations. In each map, the dark blue area represents the entire single-cell components, whereas each multi-cell component is portrayed using a different color. Quantitative information about number of single-cell components and size of the largest multi-cell component at each iteration is displayed in Table 4.1. The numbers in the table are normalized by the total number of grid cells. It can be inferred that at each iteration, a significant part of the domain is composed of single-cell components for which we update saturation by solving scalar transport equations. Moreover, multi-cell components are generally small in size, leading to small linear systems of equations for simultaneous updates of pressure and saturation. This implies that the computational cost associated with each iteration of ACHM is much lower than that of the fully coupled FIM.

Figure 4.9 shows the comparison between the fully coupled FIM with flux-based safeguard and ACHM, in terms of number of iterations required for convergence versus timestep size. For most timesteps ACHM requires fewer or the same number of nonlinear iterations. Nonetheless, as compared to the FIIM, the computational efficiency gain of ACHM for large timesteps is mostly due to the lower cost associated with each iteration rather than the number of nonlinear iterations required for convergence.

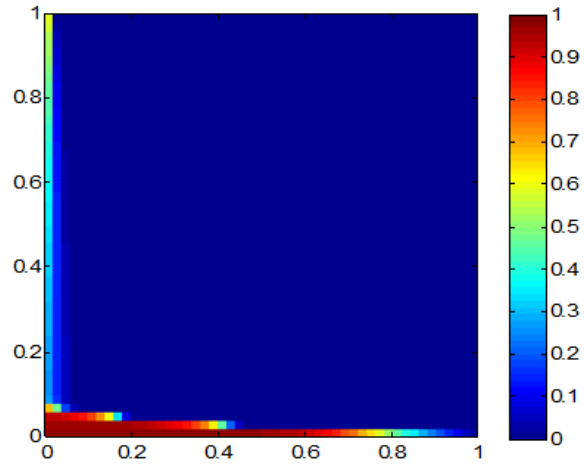


Figure 4.7: The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.

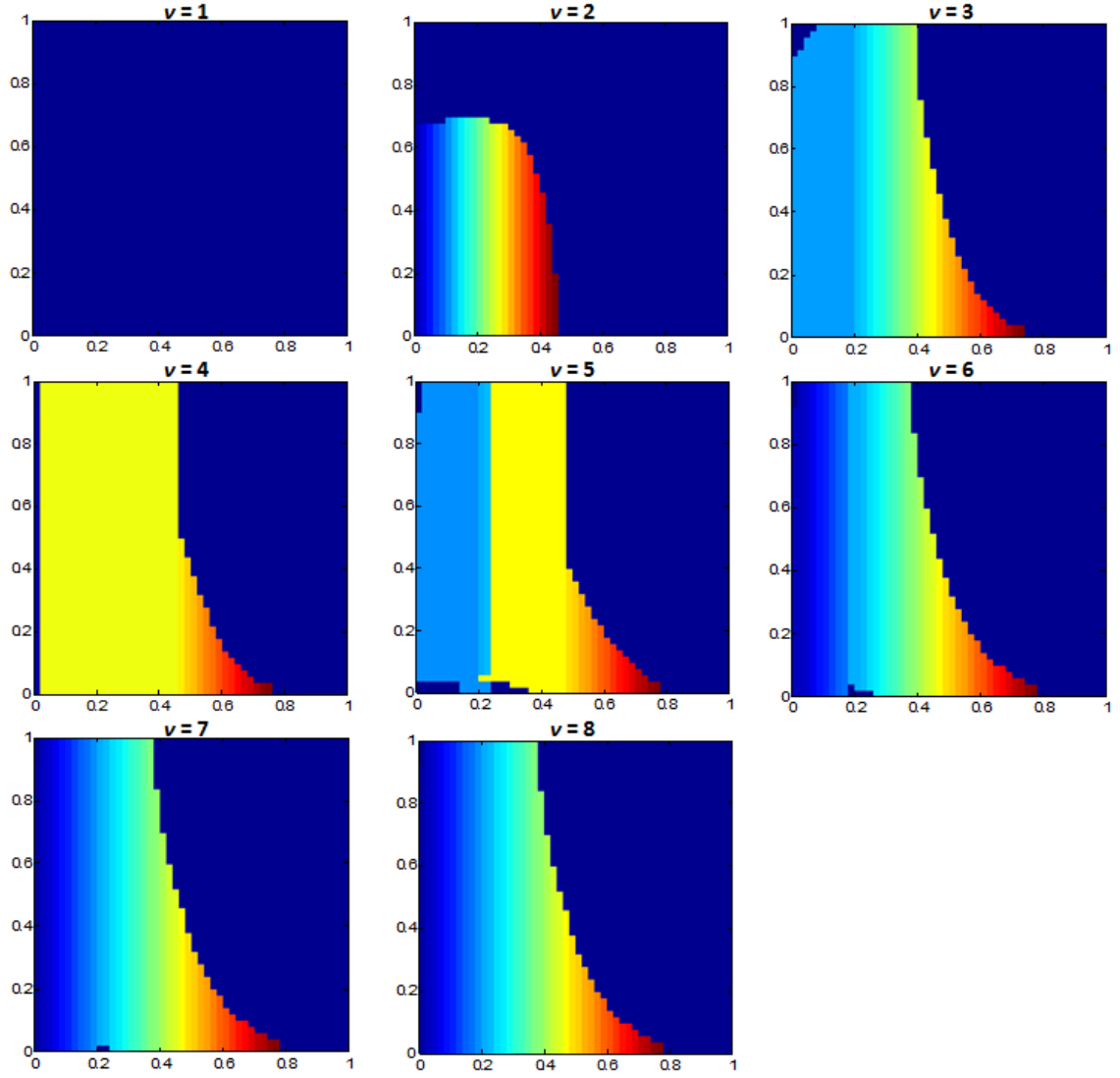


Figure 4.8: The figure shows maps of strongly connected components corresponding to different iterations. In each map, the dark blue area represents the entire single-cell components, whereas each multi-cell component is portrayed using a different color. At each iteration, a significant part of the domain is composed of single-cell components and multi-cell components are generally small in size.

Iteration	No. single-cell comp.	Size of the largest comp.
1	1.00	≤ 0.01
2	0.71	0.014
3	0.52	0.19
4	0.48	0.44
5	0.47	0.24
6	0.51	0.04
7	0.51	0.02
8	0.51	0.02

Table 4.1: Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 2500).

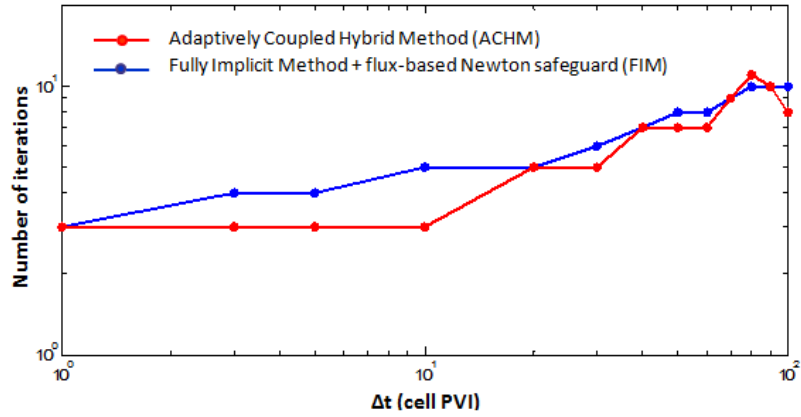


Figure 4.9: Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.

4.4.2 Heterogeneous 2D reservoir with gravity - uniform initial saturation condition

In the second example, we investigate the performance of ACHM on the 2D quarter 5-spot reservoir with a log-normal heterogeneous permeability field shown in Figure 4.10. Well locations and boundary conditions as well as fluid properties and number of cells in x , y and z directions are similar to the previous example. Saturation map at the end of one timestep, $\Delta t = 100$ cell PVI, is shown in Figure 4.11. For this large timestep, convergence is obtained after 9 iterations using ACHM.

Normalized values of number of single-cell components and size of the largest multi-cell component at each iteration are presented in Table 4.2. The number of single-cell components at each iteration is comparable to the homogeneous case. Multi-cell components are small in size and the largest component is made up of less than 10% of the entire grid cells, giving rise to less computationally demanding iterations as compared to the fully coupled FIM. Figure 4.12 shows the map of strongly connected components at convergence, i.e. iteration $\nu = 9$.

The comparison between convergence behavior of the fully coupled FIM with the flux-based safeguard and ACHM is shown in Figure 4.13. For the majority of timesteps, the number of nonlinear iterations required for convergence of ACHM is lower than that of the standard Newton with the flux-based safeguard, resulting in additional computational efficiency gain for ACHM as compared to the fully coupled FIM.

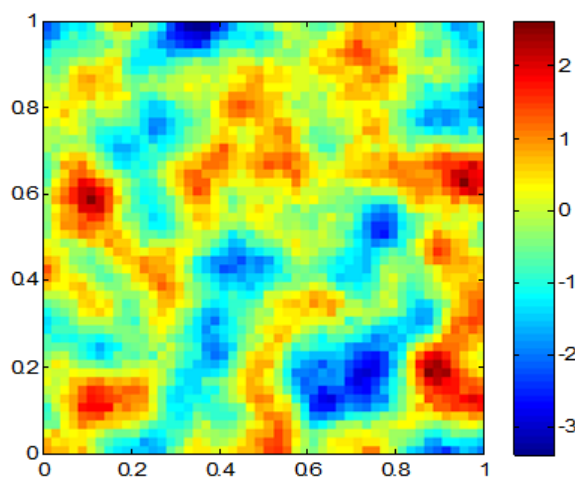


Figure 4.10: The log-normal heterogeneous permeability field used for testing ACHM.

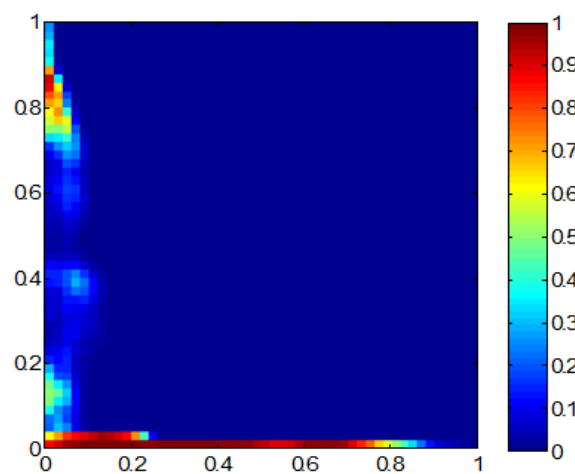


Figure 4.11: The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.

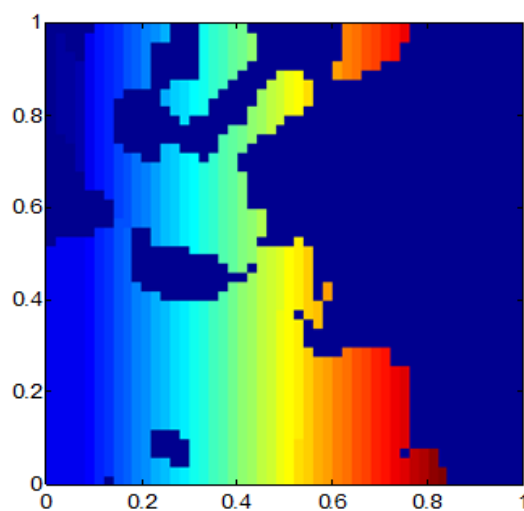


Figure 4.12: The figure shows the map of strongly connected components corresponding to iteration $\nu = 9$ of $\Delta t = 100$ cell PVI. In this map, the dark blue area represents the entire single-cell components, whereas each multi-cell component is portrayed using a different color. It can be inferred that significant part of the domain is composed of single-cell components and multi-cell components are small in size.

Iteration	No. single-cell comp.	Size of the largest comp.
1	1.00	≤ 0.01
2	0.59	0.06
3	0.51	0.11
4	0.50	0.04
5	0.50	0.10
6	0.51	0.04
7	0.51	0.04
8	0.51	0.04
9	0.51	0.04

Table 4.2: Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 2500).

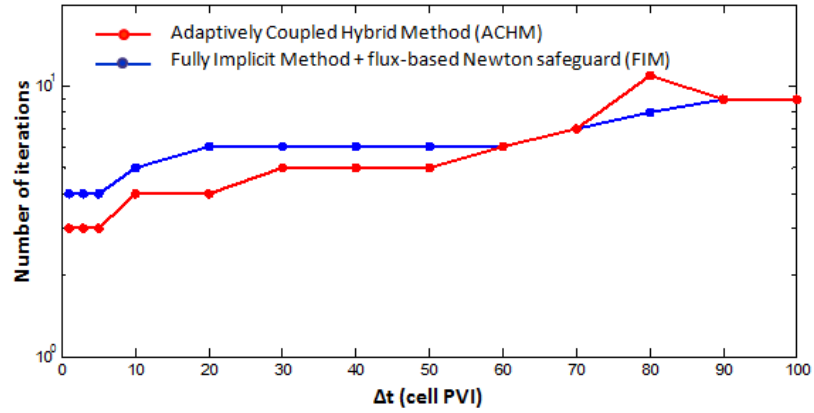


Figure 4.13: Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.

4.4.3 Top-down gravity segregation with slight injection in a heterogeneous 2D reservoir

To test the efficiency of ACHM in presence of severe counter-current flow, we change the initial saturation condition in the previous example to represent the top-down gravity segregation problem. Water is filling the upper half of the reservoir, sitting on top of oil in the lower half. Saturation map at the end of one timestep, $\Delta t = 100$ cell PVI, is shown in Figure 4.14. The redistribution of water and oil phases due to buoyancy can be observed in this figure. Convergence is obtained after 19 iterations using ACHM.

As shown in Table 4.3, At each iteration a significant part of the reservoir is composed of one large multi-cell component. Unlike previous examples, single-cell components only contribute to a small fraction of the domain. This is due to severe counter-current flow across most interfaces causing the algorithm to update pressure and saturation simultaneously in an adaptive fashion on a large fraction of the domain. Notice that as compared to the fully coupled FIM, we still have some gain in computational efficiency, since we still benefit from single-cell components to at least some degree particularly in the first 10 iterations.

Figure 4.15 shows the comparison between convergence behavior of the fully coupled FIM with the flux-based safeguard and ACHM. The number of nonlinear iterations required for convergence of ACHM is lower for most timestep sizes, leading to additional computational efficiency gain for ACHM as compared to the fully coupled FIM.

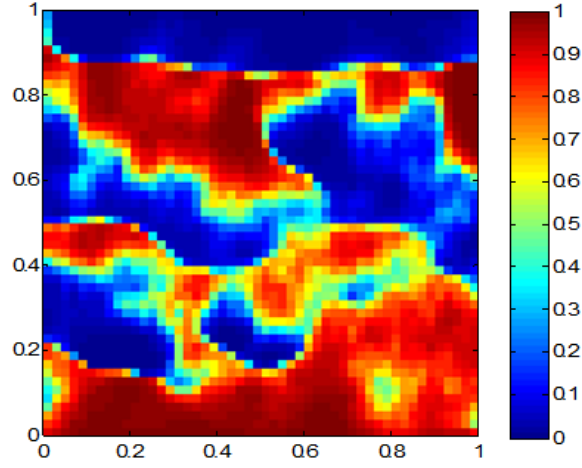


Figure 4.14: The saturation map at $t = 100$ cell PVI obtained using ACHM. The simulation was run with only one aggressive timestep, $\Delta t = 100$ cell PVI.

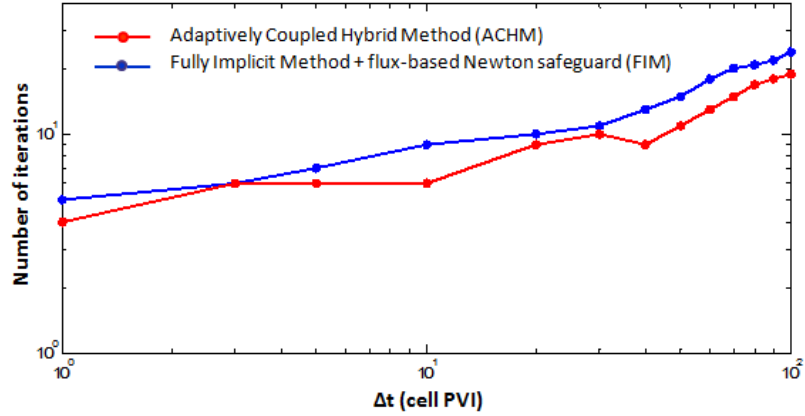


Figure 4.15: Comparison between the fully coupled FIM with flux-based safeguard and ACHM in terms of number of iterations required for convergence versus timestep size.

Iteration	No. single-cell comp.	Size of the largest comp.
1	1.00	≤ 0.01
2	0.09	0.77
3	0.05	0.92
4	0.02	0.97
5	0.01	0.85
6	0.04	0.94
7	0.02	0.95
8	0.04	0.93
9	≤ 0.01	0.98
10	≤ 0.01	0.96
11	≤ 0.01	0.98
12	≤ 0.01	0.95
13	≤ 0.01	≈ 1.00
14	≤ 0.01	≈ 1.00
15	≤ 0.01	≈ 1.00
16	≤ 0.01	≈ 1.00
17	≤ 0.01	≈ 1.00
18	≤ 0.01	≈ 1.00
19	≤ 0.01	≈ 1.00

Table 4.3: Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 2500).

4.4.4 Waterflood in a large heterogeneous 3D reservoir

To test the algorithm on a large heterogeneous reservoir, we consider a waterflood problem on a $2 \times 2 \times 2$ upscaling of the SPE 10 model (Christie and Blunt (2001)). The model has $30 \times 110 \times 42$ (a total of 138600) cells in x , y and z directions respectively. The top part of the model is a Tarbert formation and has highly variable permeabilities. The lower part (Upper Ness) is highly channelized fluvial. The upside down display of the reservoir is presented in Figure 4.16, where the channels in the bottom layers are visible. The initial saturations of oil and water are 0.8 and 0.2 respectively. Rock and fluids are incompressible. Two injectors and two producers are located at the four corners of the model and are perforated vertically through the entire layers. Water is injected at the rate of 5000 bbl/day and the producers operate at a bottom hole pressure of 4000 psi. The simulation is run up to $t = 500$ days. The initial timestep is chosen 0.05 days and subsequent timesteps are computed by the simulator based on the previous timesteps and the maximum tolerable pressure and saturation change. To avoid timestep cuts in the case of standard Newton, maximum timestep size of 50 days is considered.

Table 4.4 compares the number of iterations required for convergence, for the fully coupled FIM with safeguard and ACHM. For most timesteps, ACHM needs fewer or the same number of iterations. Nevertheless, ACHM offers tremendous computational efficiency gain due to the lower computational cost per iteration. Table 4.5 presents information about the size of various connected components at different iterations, corresponding to the final timestep ($\Delta t = 50$ days). At each iteration, approximately half of the cells form single-cell components and the size of the largest multi-cell component does not exceed 10% of the whole reservoir.

Δt (Days)	FIM iterations	ACHM iterations
0.05	4	3
0.05	3	3
0.11	4	3
0.22	5	4
0.46	5	4
0.93	5	4
1.93	6	4
4.01	6	5
8.11	6	6
15.95	7	7
50.00	9	7
50.00	8	7
50.00	10	7
50.00	7	7
50.00	7	9
50.00	7	8
50.00	7	7

Table 4.4: Comparison between the number of iterations required for convergence of the fully coupled FIM with safeguard and ACHM. For most timesteps, ACHM needs fewer or the same number of iterations.

iteration	No. single-cell comp.	Size of the largest comp.
1	1.00	≤ 0.01
2	0.55	0.10
3	0.51	0.09
4	0.52	0.07
5	0.48	0.08
6	0.49	0.07
7	0.47	0.07
8	0.47	0.07
9	0.47	0.07

Table 4.5: Number of single-cell components and size of the largest multi-cell component at different iterations (normalized by the total number of grid cells, i.e. 138600), corresponding to the final timestep, $\Delta t = 50$ days.

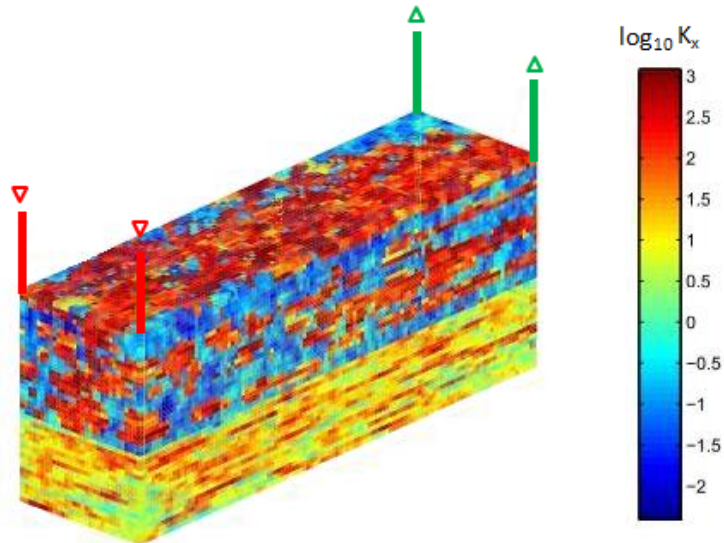


Figure 4.16: Permeability field and well configuration for the $2 \times 2 \times 2$ upscaled SPE 10 model.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

Efficient and robust nonlinear solvers are vital to the simulation of immiscible multiphase flow in porous media. Standard Newton’s method, used for solving the nonlinear equations arising from fully coupled-fully implicit discretization of flow and transport equations, fails to converge for large timesteps, especially in the presence of strong counter-current flow due to gravity. Moreover, the computational cost of each Newton iteration in a fully coupled FIM, is high due to the large size of the linear system that needs to be solved. These limitations are the main targets of the algorithms discussed in the thesis.

The trust region-based Newton safeguard proposed in Chapter 2, addresses the convergence issues of standard Newton’s method for large timestep for gravity-dominated flows. Heuristic approaches, such as Appleyard chop algorithm, improve the convergence behavior of standard Newton’s method. Nonetheless, these heuristics tend to blindly chop saturation changes on a cell-by-cell basis, and are incapable of associating the existing nonlinearities with physics of the problem. On the other hand, there is evidence that nonlinear flux function plays the key role in nonlinearity of the

multiphase flow equations. As demonstrated in various examples, convergence can be significantly improved by monitoring the saturation iterates and forcing them to stay (or barely cross) the trust regions of flux function. We show that in the case of flux with gravity, trust regions are defined by the inflection and sonic points of the flux function. Convergence maps and numerical results demonstrate the effectiveness of the proposed algorithm.

The numerical examples in Chapter 3 demonstrate the difference between the rate of convergence of pressure and saturation variables. We show that even though pressure tends to converge faster than saturation, it is vital to update both the pressure and saturation throughout the course of Newton iterations consistently in every grid cell. We also present the limitations of the reduced Newton algorithm based on phase-based potential ordering for problems with strong counter-current flow. We discuss that phase flow reversal (phase flip flop), gives rise to a mismatch between the upstream directions used to solve the pressure equation, and the ordering based on which we solve scalar saturation equations. The inability of the reduced Newton algorithm to address the coupling between pressure and saturation in the presence of gravity is presented.

To resolve the coupling between flow and transport in an efficient way, we propose the Adaptively Coupled Hybrid Method (ACHM) in Chapter 4 (Shahvali and Tchelepi (2013)). At each Newton iteration, we solve for pressure using a reduced system obtained from Schur complement reduction of the full system of equations. We then build the phase flow graph using grid cells as vertices and individual phase velocities between cells as edges. The choice of phase velocities for building the phase flow graph, rather than total velocity is advantageous in the following way. The graph obtained using the total velocity directions as edges, is likely to have cycles (strongly connected components) arising from heterogeneity of the permeability field, even for co-current flow case in the absence of gravity (Natvig et al. (2006) and Natvig and Lie (2008)).

However, the graph obtained using the phase velocity directions between the cells as edges, has no cycles in the case of co-current flow, regardless of the heterogeneity of the permeability field. On the other hand, the appearance of cycles in the phase-based graph is an indication of a local counter-current flow region. We identify the counter-current flow regions by decomposing the phase flow graph into its strongly connected components and constructing the corresponding directed acyclic graph. To update saturation variables, the components are traversed sequentially from upstream to downstream. In the case of a multi-cell component, we update saturation and pressure simultaneously to resolve the tight coupling between flow and transport due to counter-current flow. However, we update saturation nonlinearly on single-cell components that exhibit loose coupling due to co-current flow. Moreover, we safeguard the saturation iterates, for both single-cell and multi-cell components, using the algorithm proposed in Chapter 2. We present various numerical examples in 2D and 3D and demonstrate that for most problems, a significant fraction of components at each iteration are made up of single cells, giving rise to tremendous computational efficiency. Moreover, the multi-cell components are usually made up of a small number of cells as compared to the entire domain, which means the corresponding linear systems that need to be solved are small. As compared to fully coupled FIM, where we solve a large linear system to update pressure and saturation simultaneously, ACHM offers a significant computational efficiency gain.

5.2 Future directions

In the following we briefly describe a couple of possible future research directions related to this work.

5.2.1 Extending ACHM to three-phase models

The proposed nonlinear solver is generally applicable to three-phase models. More work is however required to fully understand the effects of phase compressibility as well as phase appearance and disappearance in the context of the proposed algorithm.

5.2.2 Extending ACHM to compositional models

The extension of the proposed nonlinear solver to compositional models could potentially improve the efficiency of simulations to a considerable degree. Using an implicit pressure and saturation, explicit compositions, IMPSAT formulation, the nonlinear equations in a compositional model look like a black-oil system. Phase appearance and disappearance as well as phase compressibilities should be addressed properly.

Bibliography

- Aarnes, J. E., 2004. 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.
- Appleyard, J. R., Cheshire, I. M., 1982. The cascade method for accelerated convergence in implicit simulators. In *European Petroleum Conference*, 259–290.
- Avatsmark, I., 2002. An introduction to multipoint flux approximations for quadrilateral grids. *J. Comput. Geosci.* 6, 405–432.
- Aziz, K., Settari, A., 1979. *Petroleum reservoir simulation*. Applied science publishers, New York.
- Brenier, Y., Jaffre, J., 1991. Upstream differencing for multiphase flow in reservoir simulation. *SIAM J. Numer. Anal.* 28(3), 685–696.
- Cao, H., June 2002. Development of techniques for general purpose simulators. Ph.D. thesis, Stanford University.
- Christie, M. A., Blunt, M. J., 2001. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.* 4(4), 308–317.
- Coats, K. H., September 2000. A note on impes and some impes-based simulation models. *SPE J.* 5(3), 245–251.

- Coats, K. H., George, W. D., Markum, B. E., 1974. Three-dimensional simulation of steamflooding. *Trans. SPE of AIME* 257, 573–592.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009. Introduction to algorithms. The MIT Press.
- Deuffhard, P., 2004. methods for nonlinear problems: Affine invariance and adaptive algorithms. Springer-Verlag.
- Dogru, A. H., Shaalan, L. S. K., Middy, T. M., Pita, J. A., 2008. From mega to giga-cell simulation. *SPE Annual Technical Conference and Exhibition*, Denver, Colorado.
- Edwards, M. G., 2002. Unstructured, control-volume distributed, full-tensor finite-volume schemes with flow based grids. *J. Comput. Geosci.* 6, 433–452.
- Edwards, M. G., Rogers, C. F., 1998. Finite-volume discretization with imposed flux continuity for the general tensor pressure equation. *J. Comput. Geosci.* 2, 259–290.
- Fayers, F. J., Matthews, J. D., 1984. Evaluation of normalized stone’s methods for estimating three-phase relative permeabilities. *SPE J.* 24, 224–232.
- Forsyth, P. A., Sammon, P. H., 1986. Practical considerations for adaptive implicit methods in reservoir simulation. *J. Comput. Phys.* 62, 265–281.
- Geoquest, 2005. Eclipse technical description, 2005A. Schlumberger.
- Hou, T. Y., Wu, X. H., 1997. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.* 134, 169–189.
- J. E. Dennis, J. M. M., Zhang, X., 1994. Triangular decomposition methods for solving reducible nonlinear system of equations. *SIAM J. Optimization* 4, 358–382.

- Jenny, P., Lee, S. H., Tchelepi, H., 2005. Adaptive multi-scale finite volume method for multi-phase flow and transport. *Multi-scale Model. Simul.* 3, 50–64.
- Jenny, P., Tchelepi, H., Lee, S. H., July 2009. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. *J. Comput. Phys.* 228, 7497–7512.
- Kwok, F., June 2007. Scalable linear and nonlinear algorithms for multiphase flow in porous media. Ph.D. thesis, Stanford University.
- Kwok, F., Tchelepi, H., November 2007. Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media. *J. Comput. Phys.* 227, 706–727.
- LeVeque, R. J., 1992. Numerical methods for conservation laws. Birkhuser.
- Lunati, I., Jenny, P., 2007. Multi-scale finite volume method for density driven flow in porous media. *J. Comput. Geosci.*, doi 10.1007/s10596-007-9071-9.
- McDonald, R. C., Coats, K. H., 1970. Methods for numerical simulation of water and gas coning. *Trans. SPE of AIME* 249, 425–436.
- Naccache, P. F., 1997. A fully-implicit thermal reservoir simulator. paper SPE 37985, presented at the SPE Symposium on Reservoir Simulation, Dallas, TX.
- Natvig, J. R., Lie, K. A., 2008. Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements. *J. Comput. Phys.* 227(24), 10108–10124.
- Natvig, J. R., Lie, K. A., Eikemo, B., 2006. Fast solvers for flow in porous media based on discontinuous galerkin methods and optimal reordering. *Computational methods in water resources XVI*.

- Peaceman, D. W., 1977. A nonlinear stability analysis for difference equations using semi-implicit mobility. SPE J. 17, 79–91.
- Russell, T. F., 1989. Stability analysis and switching criteria for adaptive implicit methods based on the cfl conditions. paper SPE 18416, presented at the SPE Symposium on Reservoir Simulation, Houston, TX.
- Settari, A., Aziz, K., 1975. Treatment of nonlinear terms in the numerical simulation of partial differential solutions for multiphase flow in porous media. Int. J. Multiphase Flow 1, 817–844.
- Shahvali, M., Tchelepi, H. A., 2013. Efficient coupling for nonlinear multiphase flow with strong gravity. paper SPE 163659, to be presented at the SPE Symposium on Reservoir Simulation, Woodlands, TX.
- Spillette, A. G., Hillestad, J. G., Stone, H. L., 1973. A high-stability sequential solution approach to reservoir simulation. paper SPE 4542 presented at the Fall meeting of the Society of Petroleum Engineers of AIME in Las Vegas, NV.
- Thomas, G. W., Trunau, D. H., 1983. Reservoir simulation using an adaptive implicit method. SPE J. 23, 759–768.
- Young, L. C., 1981. A finite element method for reservoir simulation. SPE J. 21(1), 115–128.
- Young, L. C., Stephenson, R. E., 1983. A generalized compositional approach for reservoir simulation. SPE J. 23(5), 727–742.
- Younis, R., August 2011. Modern advances in software and solution algorithms for reservoir simulation. Ph.D. thesis, Stanford University.

- Younis, R. M., Tchelepi, H. A., 2009. Adaptively-Localized-Continuation-Newton; reservoir simulation nonlinear solvers that converge all the time. paper SPE 119147, presented at the SPE Symposium on Reservoir Simulation, Woodlands, TX.
- Zhou, Y., November 2012. Parallel general-purpose reservoir simulation with coupled reservoir models and multi-segment wells. Ph.D. thesis, Stanford University.