THE UNIVERSITY OF TULSA
THE GRADUATE SCHOOL

LOCALIZED LINEAR SYSTEMS IN IMPLICIT TIME STEPPING FOR ADVECTION-DIFFUSION-REACTION EQUATIONS

by
Soham M. Sheth

A dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
in the Discipline of Petroleum Engineering

The Graduate School
The University of Tulsa

2018

T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

LOCALIZED LINEAR SYSTEMS IN IMPLICIT TIME STEPPING FOR ADVECTION-DIFFUSION-REACTION EQUATIONS

by
Soham M. Sheth

A DISSERTATION

APPROVED FOR THE DISCIPLINE OF

PETROLEUM ENGINEERING

By Dissertation Committee

Rami M. Younis, Chair
Mustafa Onur
Mohan Kelkar
Brett McKinney
Tom Jönsthövel

COPYRIGHT STATEMENT

ABSTRACT

Soham M. Sheth (Doctor of Philosophy in Petroleum Engineering)

Localized Linear Systems in Implicit Time Stepping for Advection-Diffusion-Reaction Equations

Directed by Rami M. Younis

141 pp., Chapter 6: Conclusions and future work

(512 words)

Implicit reservoir simulation models offer improved robustness compared to semi-implicit or explicit alternatives. The implicit treatment gives rise to a large nonlinear algebraic system of equations that must be solved at each time-step. Newton-like iterative methods are often employed in order to solve these nonlinear systems. At each nonlinear iteration, large, sparse linear systems must be solved to obtain the Newton update vector. It is observed that these computed Newton updates are often sparse, even though the sum of the Newton updates over a converged time-step may not be. Sparsity in the Newton update suggests the presence of a spatially localized propagation of corrections along the nonlinear iteration sequence. Substantial computational savings may be realized by restricting the linear solution process to obtain only the nonzero update elements. This requires an *a priori* identification of the set of nonzero update elements. To preserve the convergence behavior of the original Newton-like process, it is necessary to avoid missing any nonzero element in the identification procedure. This ensures that the localized and full linear computations result in the same solution. As a first step towards the development of such a localization method for general fully-implicit simulation, the focus is on sequential-implicit methods for general two-phase flow. We develop a mathematically sound framework to predict this sparsity pattern before the system is solved. The development first mathematically relates the Newton

update in functional space to that of the discrete system. Next, the Newton update formula in functional space is homogenized and solved in such a way that it results in conservative estimates of the numerical Newton update. The cost of evaluating the estimates is linear in the number of nonzero components. Upon projection onto the discrete mesh, the analytical estimates produce a conservative indication on the update's sparsity pattern. The estimates are used to label the components of the solution vector that will be nonzero, and the corresponding submatrix is solved. The computed result is guaranteed to be identical to the one obtained by solving the entire system.

When applied to various simulations of two- and three-phase flow recovery processes in the full SPE 10 geological model, the observed reduction in computational effort ranges between four to tenfold depending on the level of total compressibility in the system and on the time step size. We propose, apply, and test a novel algorithm to resolve a system of hyperbolic equations obtained from an Equation of State (EOS) based compositional simulator. When applied to various fully-implicit flow and multicomponent transport simulations, involving six thermodynamic species, on the full SPE 10 geological model, the observed reduction in computational effort ranges between four to twelvefold depending on the level of locality present in the system. We apply this algorithm to several injection and depletion scenarios with and without gravity to investigate the adaptivity and robustness of the proposed method to the underlying heterogeneity and complexity. We demonstrate that the algorithm enables efficient and robust full-resolution fully-implicit simulation without the errors introduced by adaptive discretization methods or the stability concerns of semi-implicit approaches.

ACKNOWLEDGEMENTS

To my life-coach, my father Monishi Sheth: because I owe it all to you. Many thanks!

To my superhero *'Omnipresent Okidanokh'*, my mother Namita Sheth, for without you nothing really makes sense. For your selfless love and all the sacrifices you made for us, I am most indebted.

I would like to express sincere gratitude to my supervisor, Rami Younis, whose selfless time and care were sometimes all that kept me going. Thank you for your resounding advice from the mayonnaise jar: "keep cool but do not freeze".

A very special gratitude goes out to Mohan Kelkar, Arthur Moncorgé, Mustafa Onur, Tom Jönsthövel and Brett McKinney for their constructive criticism and unconditional encouragement. I would also like to thank the administrative staff at The University of Tulsa, Loreta Watkins, and Rhonda Collier for their unfailing support and assistance.

I am indebted to my colleagues, Giorgiy Lutidze, Mohammadreza Mohammadnia Firoozabad, Cíntia Machado, Walter Ernesto Poquioma and Guotong Ren for countless discussions and arguments, that made this work very interesting and challenging.

I am thankful to my forever patient and encouraging wife, Heena Sheth, for her undying support and understanding in the most crucial times and for trying her best to keep me sane. I am grateful to my sister, Partha Sheth, who has provided me through much needed moral support in my life. To my new found family, Chris and Beverly Vaughn, for all our adventures together and for providing me with a home away from home. Thank you!

And finally, I would like to thank people at The University of Tulsa's Board and the FuRSST consortium for providing me with the opportunity to be a graduate assistant and for funding my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Numerous physical phenomena are modeled using nonlinear, variable-coefficient Advection-Diffusion-Reaction (ADR) systems of the form,

$$\partial_t a(u) + \partial_x f(u) + \partial_x G(u, \partial_x u) + w(u) = 0, \quad \text{in } (0,T)\times\Omega, \tag{1.0.1a}$$

$$\alpha(u) + \beta\left(\frac{\partial u}{\partial \nu}\right) = \theta, \quad \text{in } (0,T)\times\partial\Omega, \tag{1.0.1b}$$

$$u = u_0, \quad \text{for } t = 0 \tag{1.0.1c}$$

where Ω is a bounded domain in $\mathbb{R}^d$ with boundary $\partial\Omega$ and $(d = 1, 2, 3)$; $\partial u/\partial\nu$ denotes the gradient along the outward oriented unit-normal on $\partial\Omega$; $u = (u_1, \ldots, u_m)^T$ is the state vector that is comprised of the independent state variables, $u_i : \Omega \times (0,T) \to \mathbb{R}$, for $i = 1, \ldots, m$; $a(u)$ is the accumulation; $f(u)$ is the inviscid flux; $G(u, \partial_x u)$ is the viscous flux, and $w(u)$ are the reaction terms.

Individual state variables may evolve with a distinct character. For instance, in multiphase flow through porous media, the pressure state variable evolves predominantly with parabolic character, whereas the saturation state variables propagate with a finite domain of dependence, and in the limit of negligible capillary effects, as hyperbolic waves. This nonlinear superposition of characteristic spatiotemporal scales leads to rich solution structures that can undergo distinct regime changes between the parabolic and hyperbolic limiting cases. Furthermore, we are concerned with applications where generally, the nonlinear terms in Equations 1.0.1 are spatially variable with a dramatic degree of variation throughout the domain. In the numerical approximation to large scale ADR systems, the dynamic structure of the nonlinear coupling across components, and the spatial variation of the coefficients

pose a challenge to efficient and robust computation.

In terms of numerical approximations in time, explicit treatments lead to methods that are restricted by the fastest local wave speed, which due to the spatially varying coefficients and solution states, may be orders of magnitude larger than the wave speeds over the majority of the domain. Moreover, in the parabolic limit and for the diffusion terms, the stability criteria are asymptotically severe. Subsequently, explicit treatments typically lead to simulation time steps that are too conservative for the majority of the domain and solution state variables. Another challenge is in the characterization of accurate or sharp stability criteria for general nonlinear problems. Often, linear stability criteria are employed with the rationale that they will at least impose necessary conditions. In terms of computational cost, explicit methods result in fully-discrete nonlinear algebraic systems that have a block diagonal dependency structure, and that only involve mild nonlinearity. Such systems have been solved using robust methods that scale linearly with the system size. The computational expense of explicit methods is critically driven by the size of the time step that is used (e.g., [46, 24]).

Splitting methods of the implicit-explicit (IMEX) type have been proposed to overcome some of the stability requirement limitations of explicit treatments (e.g. [6, 5, 57]). In IMEX methods, the diffusive terms are typically treated implicitly, whereas advection terms are treated explicitly. While this treatment may alleviate the asymptotic severity of the stability criterion, it does not accommodate the fact that the advective local criteria may be extremely severe due to the varying coefficients. This is often the case for example in porous media flows where the permeability coefficients vary by three to five orders of magnitude throughout the domain [15]. Moreover, nonlinear stability criteria remain necessary for time step selection. On the side of computational cost, the resulting fully discrete nonlinear algebraic systems have a non-diagonal dependency structure. Large-scale preconditioned indirect solution strategies are required to compute Newton directions at each iteration of the solution process. With appropriate multigrid or domain decomposition based preconditioners, the computational cost of the linear solution process scales superlinearly with problem

size. Ultimately, IMEX approaches can be particularly effective if the underlying advective components are not too restrictive, and the nonlinearity in the flux functions is mild.

Many ADR applications in engineering and mathematical physics involve nonlinear traveling wave structures that undergo sharp local changes. This could be due to local exothermic chemical reactions for instance, or due to highly heterogeneous coefficients. In such cases, IMEX methods must observe the resulting severe stability criteria, and the corresponding time step sizes are typically too conservative for the diffusive components that are treated implicitly. Operator splitting methods have been combined with the method of fractional steps in an attempt to alleviate the computational cost requirements [52, 63, 30, 28]. In these schemes, a splitting error is introduced, and several fractional time steps may be performed on an operator by operator basis [4, 19]. For example, several fractional time steps of chemical reactions or advection may be taken per diffusion time step. The coordination and control of these schemes can be delicate. First, the splitting error that is introduced is difficult to characterize for general nonlinear problems, particularly when the operators are not additive. Moreover, the method of fractional steps introduces a need to control synchronization, leading to associated implications on accuracy and stability. When successful, operator-splitting IMEX methods with fractional steps can provide enhanced computational efficiency. Nevertheless, in highly heterogeneous cases, the spatial variation in local wave speeds remains unaddressed, requiring the solution of each operator stage to restrict fractional steps throughout the entire spatial domain according to the most locally severe criterion.

Implicit methods are a main staple of may ADR applications owing to their unconditional stability with respect to time-step size (see for example, [7, 13]). An implicit time-step requires the solution of a tightly coupled nonlinear system of discrete residual equations. One of the examples where big time-step sizes are required is the simulation of fluid flow problems in porous media. In reservoir simulation, linearized iterative solution methods such as variants of Newton's method are universally applied. In turn, each Newton-like iteration requires the solution of an equally large sparse algebraic linear system. The linear solution

process for each Newton iteration is often the dominant computational workhorse of implicit time-stepping.

In reservoir simulation, preconditioned Krylov subspace iterative methods are a popular choice for the solution of the Newton update linear systems; in particular, the Flexible General Minimal Residual (F-GMRES) method is perhaps one of the most popular Krylov methods applied. In terms of the preconditioning strategy, the goal is to strike a balance between strength (efficacy in accelerating convergence) with the setup and application cost of the preconditioner. Popular single-stage preconditioners include the classic Incomplete LU (ILU) family of methods as well as the Nested Factorization algorithm [8, 39, 59]. Given the coupling between elliptic and advective components in reservoir simulation problems, efficient multi-stage preconditioning strategies have received considerable attention in the literature. The Constrained Pressure Residual (CPR) combinative method is a two-stage preconditioner where the first stage involves the approximate solution of a reduced pressure matrix that is appropriately scaled and selected whereas the second is applied as a broad-band smoother across the entire system [58]. Options for first stage preconditioners include Algebraic Multigrid (AMG) [23, 38, 33, 53] and Algebraic Multiscale (AMS) [66, 37] methods. ILU-type methods are a popular choice for second stage preconditioners that are applied onto the updated global system. Broadly, it is observed that the computational complexity of AMG-ILU-CPR scales as $O\left(N^{\beta}\right)$ with $1.1 \leq \beta \leq 1.5$, where N is the total number of degrees of freedom in the system (see the results section for an empirical characterization of this scaling). Combined with the computational costs of evaluating the residual vector, the Jacobian matrix, and the thermodynamic property calculations, the overall complexity of a nonlinear iteration is generally super-linear in N.

Owing to the multiscale nature of coupled flow and transport, there is interest in the ability to simulate large models. The dimensionality of a simulation model scales with the required discrete spatial resolution, as well as the extent of the spatial domain. Qualitatively, extent is dictated by the fast characteristic scale of flow whereas resolution will be restricted by the sharp local character of traveling transport waves. The disparity in the

characteristic spatiotemporal scales behind flow and transport is very large, and this leads to a requirement for $N >> 1$. Recent reports on simulation models in the literature apply domains with $O(N) = 10^{12}$ (see, for example, [18]). Another consequence of the multiplicity of characteristic scales is that over short periods of time only a small fraction of the N degrees of freedom may experience considerable change. While the wide availability of parallel computing systems and algorithms enables the timely simulation of large scale models, the efficiency of the solution process may be severely limited by this fraction.

Adaptive discretization methods are designed to exploit an *a priori* knowledge, or estimates of locality to adaptively reduce the dimensionality of the discrete problem ($N_{adapt} < N$). The objective of adaptive discretization methods is to reduce the computational expense while maintaining the accuracy and fidelity of the original numerical approximation. Broadly, this is achieved by a dynamic alteration of the numerical approximation itself in order to concentrate the computational intensity onto the degrees of freedom that are undergoing significant changes. Three critical considerations are the: 1) availability of reliable forecasts that identify the rapidly evolving variables, 2) ability to reduce the required computation to a level that is proportional to their number, and 3) ability of the adapted scheme to maintain the original accuracy and stability qualities. Adaptive Mesh Refinement [25, 55, 22] methods are an example of an adaptive discretization strategy. In AMR methods, the mesh and the associated discrete approximations are adaptively coarsened to reduce the resolution in areas far away from those undergoing rapid changes. On the other hand, multirate time-stepping methods approach adaptivity by coarsening the temporal resolution locally, thereby taking several time-steps in active locales for each single coarse time-step over the remaining parts of the mesh. Owing to the severe degrees of heterogeneity and nonlinearity that are inherent to reservoir simulation problems, adaptive mesh and time-step refinement methods must address additional accuracy and robustness challenges. In particular, the dynamic homogenization of coefficients and state variables, spatial and temporal interpolation and restriction, and the adaptation algorithms must all be designed so as not to introduce additional errors while avoiding the need for excessive additional computational cost. The

Adaptive Implicit Method takes a different approach towards adaptive discretization whereby the level of implicitness of degrees of freedom are dynamically altered [47]. With adequate tuning of the criteria used to select and switch the level of implicitness, these methods may be rather successful at improving computational efficiency. As reported in [17] however, without special treatments, these methods do introduce additional errors at the interfaces between implicitly and explicitly treated variables. Moreover, an overly aggressive explicit treatment may lead to stability concerns.

While adaptive discretization methods are well studied and applied, it is conceivable that adaptive solution methods are a competitive alternative that does not alter the accuracy or stability of the approximation. In adaptive solution methods, the discretization is unaltered and can be treated fully- or adaptively-implicit. Instead, locality is exploited to improve the computational efficiency by focusing the computational intensity to the locales and variables that are undergoing change across nonlinear or linear solver iterations. While the nature of the locality that is recognized by adaptive discretization methods is limited to that across time and space, adaptive solution methods can also exploit locality across iterations. Conceivably, over a time-step, while the majority of the variables may have changed, there may be considerable locality within individual nonlinear iterations that add up to the temporal change.

1.1 Related work on adaptive solution methods

The Additive Schwartz Preconditioned Inexact Newton (ASPIN) method was proposed as a nonlinear preconditioner that is based on domain decomposition ideas [10, 27]. The ASPIN solver has been applied to solve implicit reservoir simulation time-steps [54, 50]. Locality was manifested by the need for a different number of nonlinear iterations for various subdomains. That suggested that sharp local changes, in some sense, are tied to the need for further computational effort. An outstanding technical challenge is in the effective dynamic partitioning of the domain while ensuring convergence.

In the context of nonlinear safeguarding, recent works [29, 60, 34, 40] provide nu-

merical evidence that damping the Newton updates in certain regions of the simulation grid accelerates the overall nonlinear convergence. This indirectly implies that nonlinearities evolve locally. Additionally, in [32, 41, 35] reordering methods were proposed for transport systems so that they may be lower triangularized. Subsequently, the solution process occurs sequentially in a Gauss-Seidel fashion. It was observed that some locales required more nonlinear iterations than others, suggesting a localization of nonlinearity.

In [65], it was observed that the Newton updates that are computed during the course of a Newton process are typically sparse. The authors exploit the locality within the solution process for a two-phase flow saturation Newton update to localize the linear system. The result is a direct linear solution process that is truncated to simultaneously detect locality. The algorithm uses superposition in order to isolate individual nonzero updates in the right-hand-side of the linear system. For each isolated nonzero, the linear system is then inverted directly by following the directed graph of the Jacobian matrix. Assuming that there are no directed cycles, the solutions are then combined. The authors show that the solution components decay along the directed graph, and therefore, that it is possible to terminate the process when the updates are sufficiently small. Another view of the algorithm is that it exploits the downstream ordering of saturation variables when there is no counter-current flow. The solution is then obtained by inverting the lower triangular system below the first nonzero entry in the right-hand-side. Owing to the decay property, the algorithm's worst case complexity scales linearly with the number of nonzero elements and provides the solution as well as its support simultaneously. While this approach does not degrade the overall nonlinear convergence rate, the main limitations of the algorithm are that it only applies to saturation, it requires an acyclic directed graph, and that it is particular to the first-order upwind discretization.

In [36], it is empirically observed that over the course of several Newton iterations for the solution of a time-step, the sparsity pattern of the computed Newton updates was related to that of the discrete residual vector. The authors propose to use the support set of the residual vector as an estimate for that of the Newton update. As a safety measure

they propose to inflate the support set using heuristic criteria. The inflated set is then used to solve localized systems. Because the support set is not guaranteed to be a conservative estimate in the sense described in this work, more Newton iterations are generally required using the localization procedure. In the results section, we show that severe degradation of the nonlinear convergence may be observed for typical simulation problems where ad hoc inflation criteria are insufficient. On the other hand, since the method is oblivious to the specifics of the underlying problem, it can be readily applied to simulations of complex flows.

In this work, an adaptive solution strategy is proposed for the sequential implicit simulation of general two-phase flow and transport. Similar to the methods proposed in [36] and [65], at each Newton iteration, the proposed approach identifies the subset of the degrees of freedom that will be updated over the iteration, and solves the corresponding localized linear system. The approach is mathematically conservative in that the resulting Newton updates are guaranteed to be accurate up to the linear convergence tolerance. Moreover, the proposed approach can be mathematically extended to fully-implicit simulation of general advection-diffusion-reaction systems. This is accomplished by extending the results in [64] where a sharp estimate was derived for flow in one dimension, and in [48] where general spatial heterogeneity is considered.

1.2 Outline

The proposed approach is motivated by the Asymptotic Mesh Independence Principle [62, 31, 1] that relates the convergence behavior of Newton's method for a discrete problem to that for the original continuous problem in function space. Under the assumptions of the principle, a strong connection is established between the discrete and continuous Newton iterates. In the following section, the theoretical background and basic approach are outlined. This is followed by a development of the proposed method for general sequential-implicit simulation of two phase flow in multiple dimensions. Computational results are presented that apply the proposed localization strategy, and details of future extensions to fully implicit simulation and compositional models are presented.

CHAPTER 2

THEORETICAL DEVELOPMENT

2.1 Introduction to Infinite-dimensional Newton's method

The canonical form of the nonlinear ADR PDE considered appears in Equation 1.0.1. Upon the semi-discretization in time, the independent state vector $u^{n+1} = \left(u_1^{n+1}, \ldots, u_m^{n+1}\right)^T$ is introduced, where $u_i^{n+1} : \Omega \to \mathbb{R}$, $i = 1, \ldots, m$. The solution, u^{n+1}, is the approximation to the independent state vector at discrete time level $n+1$; i.e., $u^{n+1}(x) \approx u(x, t^{n+1})$. The type of implicit discretization is unimportant so long as the resulting semi-discrete form has a single stage. Linear multistep methods follow this paradigm for example. In what follows we drop the superscript indicating the discrete time level, and we assume that all variables are at the $n+1$ level. The corresponding canonical form of the semi-discrete equations becomes,

$$R_\infty(u) := a(u) + \partial_x f(u) + \partial_x G(u, \partial_x u) = 0, \quad \text{in } \Omega, \tag{2.1.1a}$$

$$B_\infty(u) := \alpha u + \beta \frac{\partial u}{\partial \nu} = \theta, \quad \text{in } \partial\Omega, \tag{2.1.1b}$$

where $a(u)$ incorporates both the reaction and discrete accumulation. The dependence on the approximations at previous time levels is implied by the fact that all terms in Equations 2.1.1 are spatially variable. Equation 2.1.1 is a nonlinear system of PDE and is referred to as the *infinite-dimensional problem*. The residual R_∞ can be regarded as a nonlinear operator between two real Banach spaces E and $\hat{E}$; i.e. $R_\infty : W^* \subset E \to \hat{E}$.

2.1.1 Spatial discretization

The fully discrete form of the infinite problem arises upon the discretization of the

domain, the state variable, and the differential operators. This may be accomplished by several alternate discretization schemes. Let E_h and $\hat{E}_h$ denote finite-dimensional real spaces indexed by some real number $h > 0$ that denotes the mesh refinement level. Then for $U_h \in E_h$, the discretization may be described using a family of triplets,

$$\left\{R_h, I_\infty^h, \hat{I}_\infty^h\right\} \tag{2.1.2}$$

where,

$$I_\infty^h : E \to E_h, \text{ and } \hat{I}_\infty^h : \hat{E} \to \hat{E}_h,$$

are bounded linear discretization operators for a given mesh refinement $h > 0$, such that

$$I_\infty^h(W^*) \subset E_h,$$

and the fully-discrete residual is the nonlinear operator $R_h : E_h \to \hat{E}_h$ between finite-dimensional spaces. Subsequently, the resulting discrete form arises;

$$R_h\left(U_h\right) = 0. \tag{2.1.3}$$

Equation 2.1.3 is referred to as a *finite-dimensional problem* and it represents the fully discrete nonlinear residual system that must be solved at the given time step. Computationally, the finite-dimensional solution is an approximation that is hopefully accurate in space. That is, the infinite- and finite-dimensional problems are related through the spatial discretization error. For many discretizations it is shown that the finite-dimensional problem is accurate in the sense that,

$$\left\|I_\infty^h u\left(x\right) - U_h\right\| \in O\left(h^p\right), \tag{2.1.4}$$

where $p \geq 1$ is the degree of accuracy. This is a strong condition to require of discretizations. Weaker consistency-type assumptions will be used in a following section to make a connection

between the infinite and finite problems in terms of their Newton updates.

2.1.2 Linearization

Assuming Fréchet differentiability (denoted R'_∞), and invertibility of the derivative, Newton's method can be defined in infinite dimensions. Introducing the iteration index $\nu = 0, \ldots$, the generated iterates are $u^\nu(x) \in W^* \subset E$. In this infinite form, Newton's method may be applied to solve Equation 2.1.1. Starting from an initial guess, u^0, the infinite-dimensional Newton updates,

$$\delta^\nu_\infty(x) := u^{\nu+1} - u^\nu, \quad \nu = 1, \ldots,$$

are obtained by solving the linear PDE,

$$R'_\infty(u^\nu)\,\delta^\nu_\infty + R_\infty(u^\nu) = 0, \quad \text{in } \Omega, \tag{2.1.5a}$$

$$\alpha\delta^\nu_\infty + \beta\frac{\partial}{\partial\nu}\delta^\nu_\infty = 0, \quad \text{in } \partial\Omega, \tag{2.1.5b}$$

Equations 2.1.5 are referred to as the *infinite-dimensional Newton problem*. Notice that by the linearity of the boundary value conditions in Equations 2.1.1 the resulting boundary conditions for the Newton problem are homogeneous, assuming that the initial guess satisfies the boundary conditions and that the conditions are constant in time.

Analogously, assuming that the finite-dimensional residual equations are differentiable, and that the resulting Jacobian matrix is invertible, the finite-dimensional problem may be solved using Newton's method as well. The iteration is started from $U^0_h := I^h_\infty u^0(x)$, and the update directions are obtained by solving the linear algebraic system,

$$R'_h(U^\nu_h)(\delta^\nu_h) + R_h(U^\nu_h) = 0 \;\; \nu = 1, \ldots. \tag{2.1.6}$$

The basic premise of the work proposed relies on a connection between δ^ν_h and δ^ν_∞. This connection is developed next.

2.1.3 A connection between finite- and infinite-dimensional Newton updates

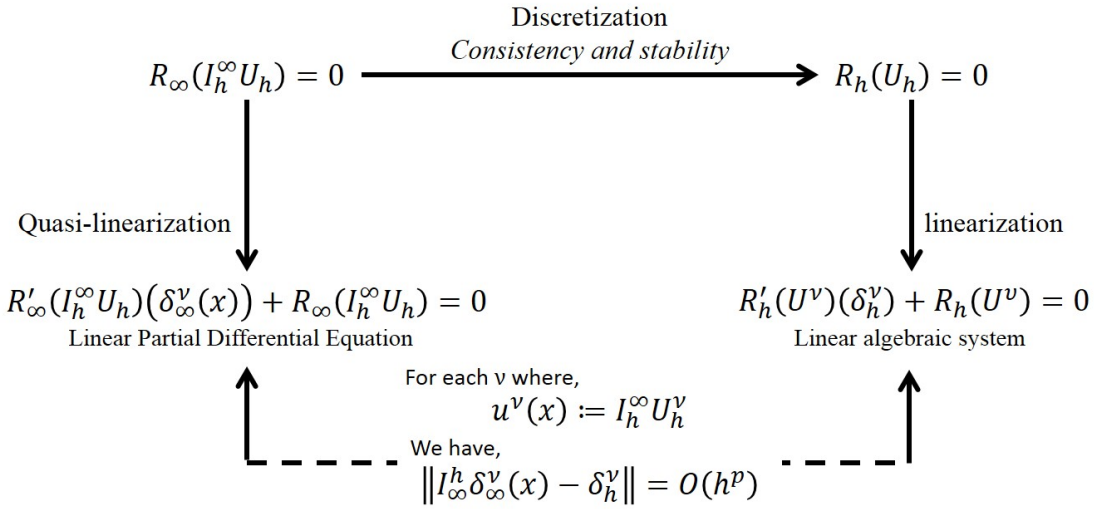


Figure 2.1: The connections between the Newton processes for the infinite- and finite-dimensional problems.

The infinite- and finite-dimensional Newton updates will be related to each other under certain assumptions on the discretization scheme. Along with the assumptions on the discretization scheme, careful consideration should be given to the continuous state variable field, $u^\nu(x)$. In this work, $u(x)$ is obtained at every iteration from the discrete state variable, U_h. Hence, for each iteration ν,

$$u^\nu(x) = I_h^\infty U_h^\nu.$$

If the initial guess is inside the contraction region, this condition is not necessary. As in most practical applications, its difficult if not impossible to predict the contraction region, this condition becomes very important for the following proofs and the application to advection-diffusion-reaction problems. This situation is summarized in Figure 2.1. We assume that the Equations 2.1.5 and 2.1.6 are well defined on $W^* \subset E$. Moreover, due to the smoothness of the Newton updates, we assume that all the iterates remain inside W^*. We also introduce the q-norm $\|\cdot\|$ on E_h and on $\hat{E}_h$ as well as the associated induced operator norm. With these assumptions, we make definitions concerning the stability and consistency of the discrete finite-dimensional problem. These definitions will allow us to establish a connection between the infinite and finite Newton updates.

Definition 1. *A discretization* $\left\{R_h, I_\infty^h, \hat{I}_\infty^h\right\}$ *defined on* $u \in W^* \subset E$ *for* $h > 0$ *is called stable if there is a constant* $\sigma > 0$ *such that*

$$\left\|R_h'\left(U_h\right)^{-1}\right\| \leq \sigma,$$

for $u \in W^* \subset E$.

Additionally, we define notions of consistency on the discrete system, thereby introducing the mesh refinement parameter $h > 0$ and an asymptotic truncation error of order $p \geq 1$.

Definition 2. *A discretization* $\left\{R_h, I_\infty^h, \hat{I}_\infty^h\right\}$ *is consistent of order* p *if there are two constants* $c_0 > 0$ *and* $c_1 > 0$ *such that*

$$\left\|\hat{I}_\infty^h R_\infty\left(I_h^\infty U_h\right) - R_h\left(U_h\right)\right\| \leq c_0 h^p,$$

and,

$$\left\|\hat{I}_\infty^h R_\infty'\left(I_h^\infty U_h\right) v - R_h'\left(U_h\right)\hat{I}_\infty^h v\right\| \leq c_1 h^p,$$

for $u, v \in W^* \subset E$.

With these definitions, we introduce Theorem 1 that constitutes a key relationship that will be exploited by the proposed work.

Theorem 1. *Given an infinite problem (Equation 2.1.1), let the discretization* $\left\{R_h, I_\infty^h, \hat{I}_\infty^h\right\}$ *for some* $h > 0$ *satisfy Definitions 1 and 2. Let* $U_h^\nu \in I_h^\infty W^* \subseteq E_h$ *and* $u^\nu = I_h^\infty U_h^\nu$. *Then, given a positive constant* c_3, *the Newton update* δ_∞^ν *that is defined by Equation 2.1.5 and* δ_h^ν *that is defined by Equation 2.1.6 satisfy,*

$$\left\|\delta_h^\nu - I_\infty^h \delta_\infty^\nu\right\| = c_3 h^p \quad \nu = 1, 2, \cdots . \tag{2.1.7}$$

Proof. Similar proofs with stronger conditions can be found in [2] and [3] as intermediate results towards developing the Asymptotic Mesh Independence Principle (AMIP). We obtain

the identity,

$$\begin{aligned}
\delta_h^\nu - I_\infty^h \delta_\infty^\nu &= -R_h'\left(U_h^\nu\right)^{-1}\left[R_h\left(U_h^\nu\right) - R_h'\left(U_h^\nu\right)\hat{I}_\infty^h R_\infty'\left(I_\infty^h U_h^\nu\right)^{-1} R_\infty\left(I_h^\infty U_h^\nu\right)\right] \\
&= -R_h'\left(U_h^\nu\right)^{-1}\left\{\left[R_h\left(U_h^\nu\right) - \hat{I}_\infty^h R_\infty\left(I_h^\infty U_h^\nu\right)\right]\right. \\
&\quad \left. - \left[R_h'\left(U_h^\nu\right)\hat{I}_\infty^h R_\infty'\left(I_h^\infty U_h^\nu\right)^{-1} R_\infty\left(I_h^\infty U_h^\nu\right) - \hat{I}_\infty^h R_\infty\left(I_h^\infty U_h^\nu\right)\right]\right\} \\
&= -R_h'\left(U_h^\nu\right)^{-1}\left\{\left[R_h\left(U_h^\nu\right) - \hat{I}_\infty^h R_\infty\left(I_h^\infty U_h^\nu\right)\right]\right. \\
&\quad - \left[R_h'\left(U_h^\nu\right)\hat{I}_\infty^h R_\infty'\left(I_h^\infty U_h^\nu\right)^{-1} R_\infty\left(I_h^\infty U_h^\nu\right)\right. \\
&\qquad \left.\left. - \hat{I}_\infty^h R_\infty'\left(I_h^\infty U_h^\nu\right) R_\infty'\left(I_h^\infty U_h^\nu\right)^{-1} R_\infty\left(I_h^\infty U_h^\nu\right)\right]\right\}
\end{aligned} \tag{2.1.8}$$

Since the infinite iterates all remain within the W^*, we have that

$$R_\infty'\left(I_h^\infty U_h^\nu\right)^{-1} R_\infty\left(I_h^\infty U_h^\nu\right) \in W^*,$$

and subsequently, we can apply the two conditions in Definition 2 to the two bracketed terms above. Using the triangle inequality we obtain,

$$\left\|\delta_h^\nu - I_\infty^h \delta_\infty^\nu\right\| = c_3 h^p.$$

□

Example 1. *To show the connection between the finite and infinite Newton updates, consider a nonlinear first order equation of the form*

$$R_\infty(u) := u(x) + \partial_x f\left(u(x)\right) = 0, \quad x \in [0,1], \tag{2.1.9a}$$

$$B_\infty(u) := u(x) = u_0, \quad x = 0. \tag{2.1.9b}$$

This equation describes the flow of a fluid in the presence of another and is commonly known as the Buckley-Leverett equation. Given a consistent and stable discretization scheme, it can

be shown that

- *The number of iterations required for the finite- and infinite-dimensional Newton's methods to converge to the solution are similar and*
- *Equation 2.1.7 is satisfied.*

Solution. The above problem is solved numerically using the finite difference discretization scheme. It is very simple to prove that finite difference approximation of a first order derivative is first order accurate. In Equation 2.1.9, $u(x)$ is a scalar state variable and for a constant $M > 0$, f is a nonlinear flux function given by

$$f = \frac{u^2}{u^2 + M(1-u)^2}.$$

Figure 2.2 shows the s-shaped fractional flow curve, along with its first and second derivative, for $u \in [0, 1]$. Equation 2.1.9 is solved for two different initial conditions as shown in Figures 2.3a and 2.4a. Case 1 is designated to the s-shaped initial condition with boundary condition at the left boundary, $u(x = 0) = 0.9735$. Case 2 is the exponentially decaying initial distribution with the left boundary value, $u(x = 0) = 0.5104$. Figures 2.3b and 2.4b show the snapshots of finite- and infinite-dimensional Newton updates for a time step size of 0.25 and 0.5 days, respectively. It can be readily seen that the number of iterations are exactly the same and at the same time, each iterate of both the methods are comparable. In theory, the number of iterations for the finite- and infinite-dimensional Newton methods can differ at most by one. This illustrates the first part of Example 1.

Along with the similarity of the two methods in terms of the iteration count, it can also be shown that the absolute values of each update differs at most by the discretization error. To prove Equation 2.1.7 holds for Equation 2.1.9, maximum absolute discretization error is computed for various refinement levels as shown in Figures 2.5a and 2.5b. The reference solution is taken to be the updates obtained from the infinite-dimensional method. As it can be seen from the figures, for each iteration, the discretization error is first order accurate. This illustrates the second part of Example 1.

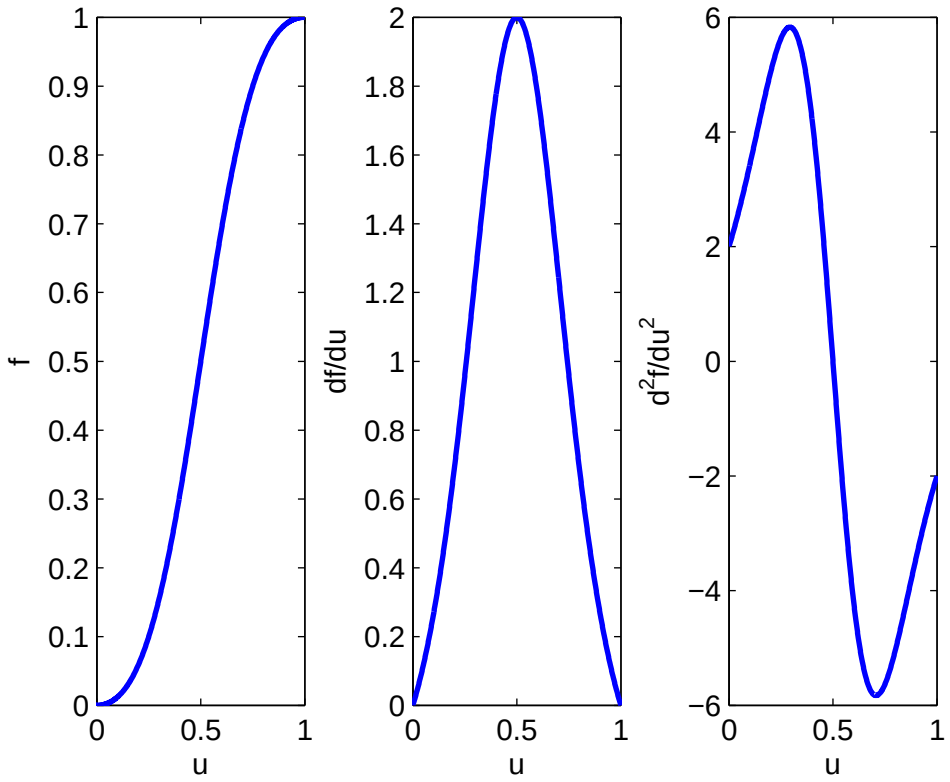


Figure 2.2: This figure shows the nonlinear s-shaped fractional flow curve and its first and second derivative for $u \in [0, 1]$.

2.2 Conservative estimate in the sense of support

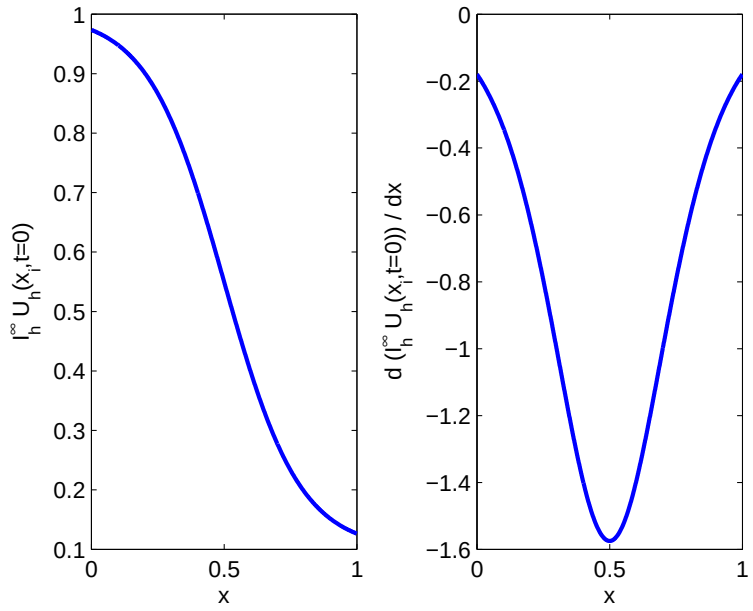
2.2.1 Non-zero support set

The *non-zero support set* for a Newton update is defined as the collection of indices for which the update component is nonzero; i.e.,

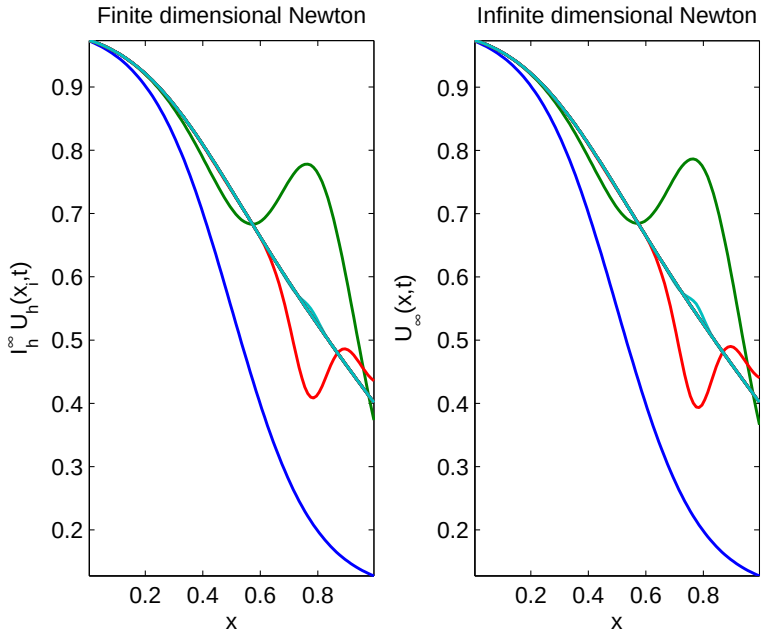
$$\operatorname{supp} \delta_h = \{i : |\delta_{h,i}| > \epsilon, i = 1, \ldots, N\}, \tag{2.2.1}$$

where $\epsilon = O(h^p)$. The zero pseudo-norm is then the cardinality of the support set; i.e.,

$$\|\delta_h\|_0 = \# \operatorname{supp} \delta_h. \tag{2.2.2}$$

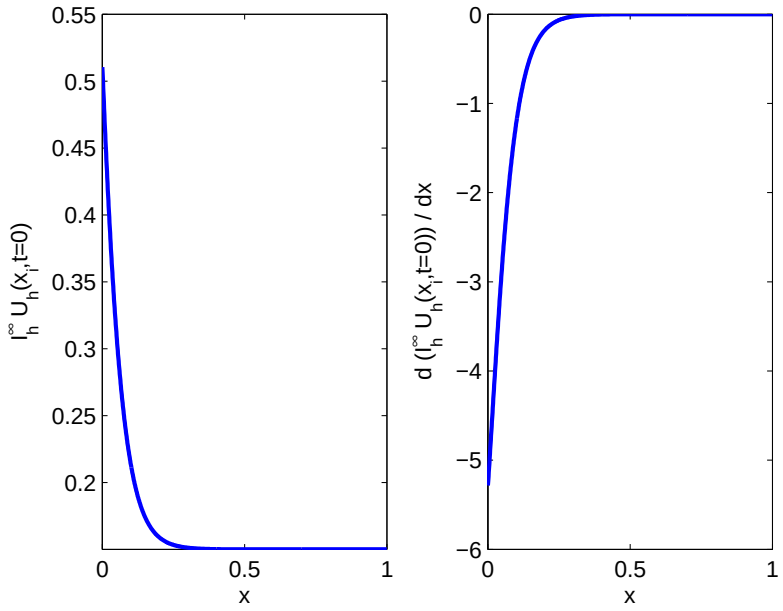


(a) Initial condition for the state variable.

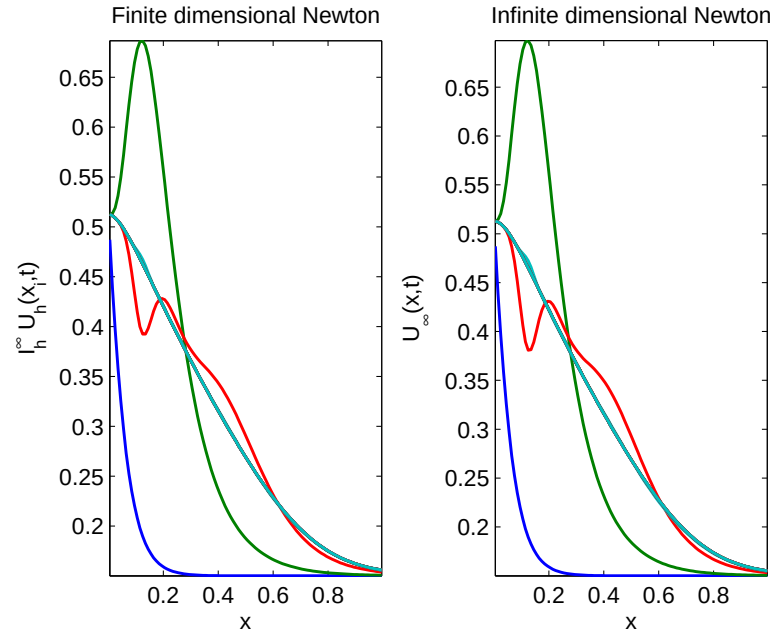


(b) Comparison of Newton updates for infinite and finite dimensional Newton methods.

Figure 2.3: Numerical results for the Asymptotic Mesh Independence Principle for a wave problem with heterogeneous initial condition and the comparison of finite dimensional Newton's method with the continuous counterpart in the Banach space. Sub-figure (a) shows the initial state and its derivative along the physical domain while sub-figure (b) shows the Newton updates obtained for consecutive iterations.

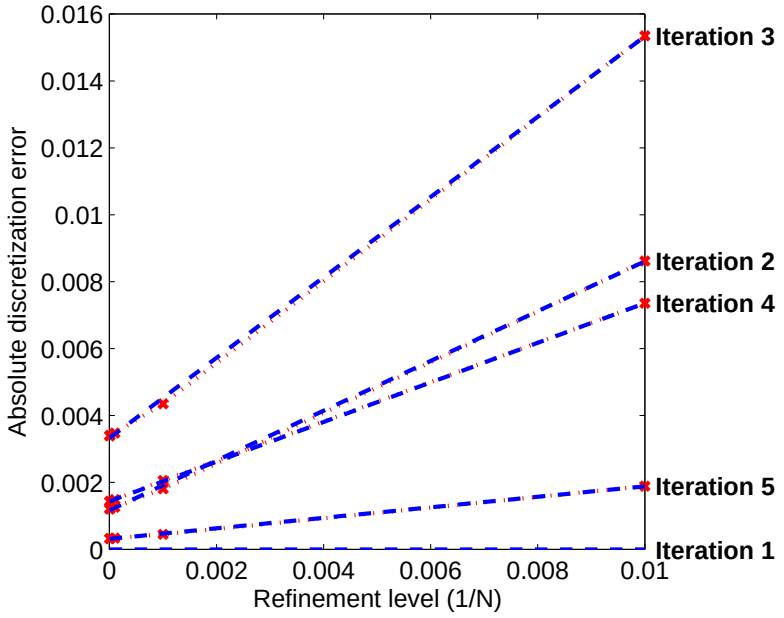


(a) Initial condition for the state variable.

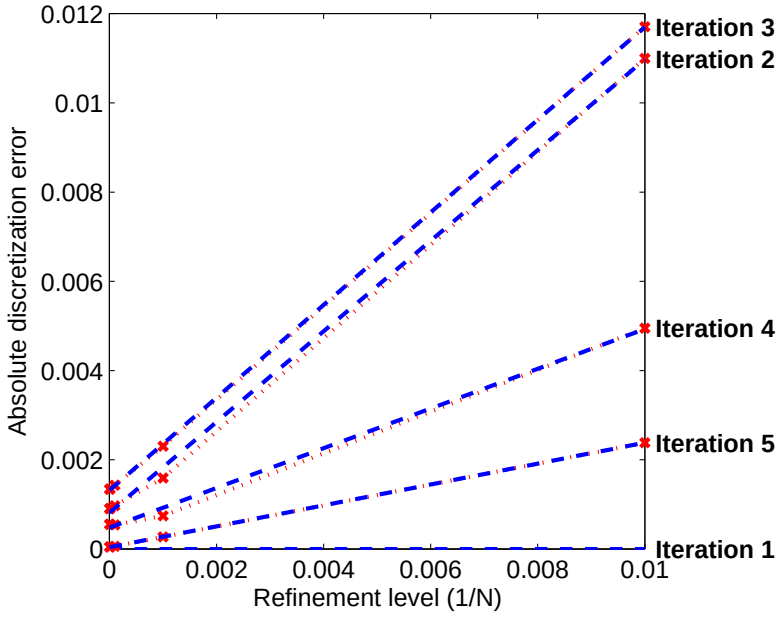


(b) Comparison of Newton updates for infinite and finite dimensional Newton methods.

Figure 2.4: Numerical results for the Asymptotic Mesh Independence Principle for a wave problem with heterogeneous initial condition and the comparison of finite dimensional Newton's method with the continuous counterpart in the Banach space. Sub-figure (a) shows the initial state and its derivative along the physical domain while sub-figure (b) shows the Newton updates obtained for consecutive iterations.



(a) Discretization error as a function of the refinement level for case 1.



(b) Discretization error as a function of the refinement level for case 2.

Figure 2.5: Sub-figures (a) and (b) are obtained by successively refining the homogeneous mesh and evaluating the infinity norm of the discretization error. The reference solution is taken to be the root of the infinite-dimensional Newton's iterate which is obtained by solving a linear Ordinary Differential Equation (ODE) with a high fidelity ODE solver. The results show the error to be first order accurate for finite difference approximation.

Given two vectors, V_h, and W_h, such that,

$$|V_h| - |W_h| \geq 0, i = 1, \ldots, N, \tag{2.2.3}$$

then the support of the first update contains the support of the second; i.e.,

$$\operatorname{supp} W_h \subseteq \operatorname{supp} V_h. \tag{2.2.4}$$

Using these facts and Theorem 1, it is easy to show the following result (Corollary 1) that characterizes the relationship between the nonzero support sets of the infinite and finite Newton updates.

Corollary 1. *Let the discretization* $\left\{R_h, I_\infty^h, \hat{I}_\infty^h\right\}$ *for some* $h > 0$ *satisfy Definitions 1 and 2 with respect to an infinite problem of the form 2.1.1 in the infinity norm* $\|\cdot\|$*. Then the Newton update,* δ_∞^ν *that is defined by Equation 2.1.5, and* δ_h^ν *that is defined by Equation 2.1.6 satisfy,*

$$\operatorname{supp} \delta_h^\nu \subseteq \operatorname{supp} I_\infty^h \delta_\infty^\nu \cup \operatorname{supp} \epsilon_d, \tag{2.2.5}$$

where $\epsilon_d = O(h^p)$ *is a form of the discretization error.*

Proof. By simple application of the triangle inequality, for any $\nu = 0, 1, \ldots,$ we have that,

$$|\delta_h^\nu| = \left|0 - I_\infty^h \delta_\infty^\nu + I_\infty^h \delta_\infty^\nu - \delta_h^\nu\right| \leq \left|I_\infty^h \delta_\infty^\nu\right| + \left|I_\infty^h \delta_\infty^\nu - \delta_h^\nu\right|.$$

But by application of Theorem 1 in the infinite norm, we have that, for each component, $i = 1, \ldots, N$,

$$\left|\delta_{h,i}^\nu\right| \leq \left|\left(I_\infty^h \delta_\infty\right)_i^\nu\right| + |\epsilon_d|, \tag{2.2.6}$$

and the claim follows directly from the definition of the support set. □

2.2.2 Localized computational methods for sparse Newton updates

Although a connection was established between infinite and finite Newton updates, there remains to be two difficulties with a direct application. First, the infinite Newton update may be impossible to obtain; it is the solution to a nonlinear ADR boundary value problem with variable coefficients and general linear boundary conditions over an arbitrary domain. Secondly, there remains the uncharacterized discretization error term that links the two updates. By introducing the Definition 3, we can develop an intermediary infinite-dimensional update that is easy to evaluate and leads to a direct connection to the finite-dimensional update.

Definition 3. *Given an infinite-dimensional Newton process as defined by Equation 2.1.5, and a corresponding update, $\delta_\infty \in W^* \subset E$, we say that $\delta^* \in W^* \subset E$ is a conservative estimate in the sense of support if there exists a constant $C_0 \geq 0$ independent of $x \in \Omega$, such that*

$$|\delta^*(x)| - |\delta_\infty(x)| = C_o \geq 0.$$

Conservative estimates in the sense of support are directly related to the finite-dimensional Newton updates for various discretizations according to the following results.

Theorem 2. *Consider a well-defined process according to Equation 2.1.5 with the corresponding discrete form as in Equation 2.1.6. Suppose that the discretization satisfies the assumptions in Definitions 1 and 2 in the infinity norm. If for a given iteration, $\nu = 1, \ldots,$ we have that $\delta^* \in W^* \subset E$ is a conservative estimate in the sense of support to δ_∞^ν, and,*

$$C_0 - \epsilon_d \geq 0,$$

then,

$$\operatorname{supp} \delta_h^\nu \subset \operatorname{supp} \left[I_\infty^h \delta^* \right]^\nu.$$

Proof. Using the triangle inequality and the fact that according to Theorem 1 $\left| \delta_h^\nu - I_\infty^h \delta_\infty^\nu \right| =$

$O\left(h^p\right)$, we have,

$$\begin{aligned}
\left|I_\infty^{h,i}\delta^*\right| - \left|\delta_{h,i}\right| &= \left|I_\infty^{h,i}\delta^*\right| - \left|\delta_{h,i} - I_\infty^{h,i}\delta_\infty + I_\infty^{h,i}\delta_\infty\right| \\
&\geq \left|I_\infty^{h,i}\delta^*\right| - \left|I_\infty^{h,i}\delta_\infty\right| - \left|\delta_{h,i} - I_\infty^{h,i}\delta_\infty\right| \\
&\geq C_0 - \left|\delta_h^\nu - I_\infty^h\delta_\infty^\nu\right| \geq 0
\end{aligned}$$

The claim follows directly from the definition of the support set. □

2.3 Objective

The key objective is the development of a theoretical basis and an associated algorithmic process for the identification of the elements of a Newton update that may be neglected, prior to solving the linear system. The key idea in this work is to accomplish this by projecting the Newton process for the discrete residual system onto a Newton process to solve a continuous counterpart. Subsequently, classical and proven ideas in homogenization theory and eigenvalue localization for linear PDE can be applied. With these simplifications, the spatial support of the continuous Newton update may be easily (analytically) estimated and projected back onto the discrete form. The main motivation for adopting this approach is that the estimates are rather independent of the particular numerical scheme that is used.

This method can be applied to any ADR equation given by Equation 2.1.1. In this work a particular problem pertaining to the fluid flow in porous media is considered. One of the classical models to simulate the flow of two or more fluids in rocks is the black oil model. The details of this model can be found in Appendix A, along with various auxiliary data. Appendix A presents the governing equations for compressible, multi-dimensional, two-phase flow with capillary, and buoyancy effects. These equations are aligned with two independent variables such as pressure and saturation. This work focuses on the Newton update computations that arise within the sequential implicit solution algorithm [51, 61]. In the sequential solution procedure for a time step, an outer-iteration is performed wherein the flow and transport equations are approximated sequentially. That is, at each outer iteration,

the flow equation is discretized implicitly in pressure, while saturation is frozen at the latest estimate. A Newton-type of procedure is used to solve this discrete approximation to obtain an updated pressure field. In the second step, the updated pressure field is frozen while the transport equation is discretized implicitly in saturation. Once again, a Newton-type process is used to obtain the updated saturation field. Ultimately, and from the perspective of each of the two Newton processes that are applied sequentially, the residual equation that is to be solved is a scalar advection-diffusion-reaction problem.

One implication of the close connection between the continuous and discrete Newton updates is that the projection of the infinite dimensional solution onto the simulation mesh produces an accurate estimate of the discrete Newton update up to an asymptotically vanishing discretization error. This error is almost always over-shadowed by other sources of error. For example, considering that indirect methods are often applied in the context of Newton-Krylov methods for instance, there is an absolute error tolerance on the discrete Newton update that is in play; i.e.,

$$\delta_h^{\text{computed}} = \delta_h + \epsilon_{\text{linear}},$$

where ϵ_{linear} can be significant. Moreover, the Newton convergence criteria often involve a stopping tolerance on the Newton update size such that if,

$$\|\delta_h^\nu\| < \epsilon_{\text{nonlinear}},$$

for some prescribed $\epsilon_{\text{nonlinear}}$, then the iteration is terminated. As a matter of practice, we will define the characteristic floating point tolerance $\epsilon > 0$ to encompass the accuracy considerations of near-zero quantities. That is, entries of the Newton update such that $|\delta_{h,i}| \leq \epsilon$ are essentially negligible and by neglecting them, the convergence behavior of this truncated Newton and its counterpart are identical.

Naturally, it is intractable to derive the infinite dimensional solution for general problems directly. If it were tractable, then numerical simulation would not have been used.

Instead, we will propose certain types of estimates that are easily obtained and that can indicate the near-nonzero degrees of freedom. To quantify the near-zero quality, the support set for a Newton update is defined as the collection of indices for which the update component is essentially nonzero:

$$\operatorname{supp} \delta_h = \{i : |\delta_{h,i}| > \epsilon, i = 1, \ldots, N\}. \tag{2.3.1}$$

The cardinality of the support set, i.e. the zero norm, is represented as,

$$\|\delta_h\|_0 = \# \operatorname{supp} \delta_h. \tag{2.3.2}$$

Given two update vectors, δ_h^1, and δ_h^2, such that,

$$\left|\delta_{h,i}^1\right| \geq \left|\delta_{h,i}^2\right|, i = 1, \ldots, N, \tag{2.3.3}$$

the support of the first update contains the support of the second,

$$\operatorname{supp} \delta_h^1 \supseteq \operatorname{supp} \delta_h^2, \tag{2.3.4}$$

and we say that δ_h^1 is a *conservative estimate for* δ_h^2 *in the sense of support*. The remainder of the theoretical development in this work concerns the development of model problems, as described in this chapter, of the form

$$\begin{cases} \hat{R}'_\infty \left(I_h^\infty U_h^\nu\right) \delta_\infty^* \left(x\right) + I_h^\infty R_h = 0 & x \in D \subset \mathbb{R}^3 \\ \hat{B}_\infty \left(\delta_\infty^* \left(x\right)\right) = 0 & x \in \partial D \end{cases}, \tag{2.3.5}$$

that are chosen to be easy to solve analytically while providing a conservative estimate $\delta_\infty^* \left(x\right)$ to the solution of problem 2.1.6. The application is to then solve a reduced linear system whose dimension is $\left\|I_\infty^h \delta_\infty^*\right\|_0 \leq N$ to obtain the full Newton update itself up to the ϵ tolerance. This is depicted schematically in Figure 2.6.

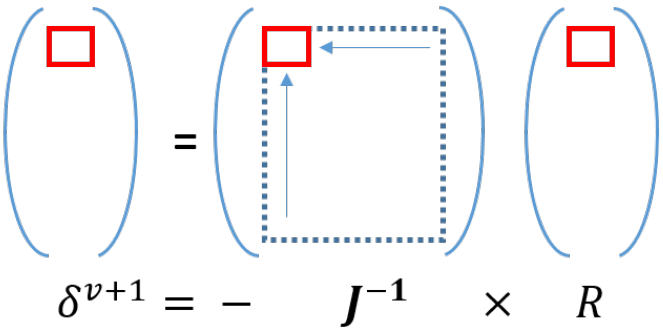


Figure 2.6: Localized linear solutions can be obtained if the support (depicted in red) of the unknown is known before hand.

CHAPTER 3

DERIVATION OF ANALYTICAL ESTIMATES

The mathematical framework developed in this work is generally applicable to the evolution of any physical state variable that fits the ADR canonical form. In the following development and analysis special focus will be given to the physics governing fluid flow processes in porous media and certain analogy will be drawn to make general derivations and conclusions.

3.1 The infinite-dimensional problem and Newton process

The point of departure for the analysis is the semi-discrete-in-time form of the governing equations for flow and transport. Semi-discrete-in-time equation is considered because the entire mathematical framework proposed in this research is pertaining to one time-step. While operating over several Newton iterations, the time-step size remains constant and hence the analysis is carried out for varying spatial variable while the time state remains frozen. Let $\Omega \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$ denote the open and bounded domain of interest with boundary $\partial\Omega$. We will consider the evolution of the state variables over a time-step $\Delta t > 0$. In this continuous-in-space setting, either of the two subproblems of interest (flow or transport) is independently aligned with a single semi-discrete state variable denoted $u : \Omega \subset \mathbb{R}^d \to D \subseteq \mathbb{R}$ (pressure or saturation). This variable is the numerical approximation to the solution of the problem at the end of the time-step. The second state variable (saturation or pressure respectively) is frozen as a fixed function of space and is absorbed into the definitions of the spatially variable coefficients in the PDE.

In this context, the general canonical semi-discrete form of Equation 1.0.1 is,

$$\begin{cases} R_\infty\left(u\left(x\right)\right) = a\left(x, u\left(x\right)\right) + \Delta t \nabla \cdot \mathbf{f}\left(x, u\left(x\right)\right) \\ \qquad\qquad + \Delta t \nabla \cdot \left[\mathbf{G}\left(x, u\left(x\right)\right) \nabla u\left(x\right)\right] + \Delta t w\left(x, u\left(x\right)\right) = 0 \quad x \in D \subset \mathbb{R}^d, \\ B_\infty\left(u\left(x\right)\right) = \alpha_1\left(x, u\left(x\right)\right) + \alpha_2\left(x, \frac{\partial u}{\partial \nu}\right) = 0 \qquad\qquad x \in \partial\Omega, \end{cases} \tag{3.1.1}$$

where $\frac{\partial u}{\partial \nu}$ is the directional gradient along the outward-oriented unit-normal on $\partial\Omega$; $a : \Omega \times D \to \mathbb{R}$ is the change in accumulation over the time-step; $\mathbf{f} : \Omega \times D \to \mathbb{R}^d$ is the inviscid flux; $\mathbf{G} : \mathbb{R}^d \to \mathbb{R}^d$ is the viscous flux coefficient operator; $w : \Omega \times D \to \mathbb{R}$ is the net sink term; and $\alpha_{1,2}$ parameterize the boundary conditions of the problem. These coefficients can be chosen appropriately for either the flow or the transport sub-problems as shown in Appendix A.

Equation 3.1.1 is referred to as the *infinite-dimensional* residual and it is a nonlinear Boundary Value Problem (BVP). Moreover, the residual R_∞ can be regarded as a nonlinear operator between the two real Banach spaces E and $\hat{E}$; i.e. $R_\infty : W^* \subset E \to \hat{E}$.

Assuming Fréchet differentiability (denoted R'_∞), and invertibility of the derivative operator, Newton's method can be defined in infinite dimensions. Note that while the coefficients in Equation 3.1.1 may very well be only piecewise differentiable in space (i.e. with respect to x), what is required for Fréchet differentiability in this context is that they be differentiable with respect to the state variable u over the time-step and not necessarily so in space.

Fréchet derivative can be interpreted as the directional derivative of R_∞ in the direction of δ_∞. Mathematically,

$$R'_\infty(u(x))\delta_\infty = \frac{d}{d\epsilon} R_\infty(u(x) + \epsilon\delta_\infty)\bigg|_{\epsilon=0}.$$

Given an initial state, u^ν, the perturbed equation can be written as

$$\begin{aligned} R_\infty\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right) =& a\left(x,\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right)\right)+\Delta t\nabla\cdot\mathbf{f}\left(x,\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right)\right) \\ &+\Delta t\nabla\cdot\left[\mathbf{G}\left(x,\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right)\right)\nabla\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right)\right] \\ &+\Delta t w\left(x,\left(u^\nu(x)+\epsilon\delta_\infty^\nu\right)\right)=0. \end{aligned} \tag{3.1.2}$$

Applying the definition of Fréchet derivative to Equation 3.1.2 and condensing the notation, we obtain

$$\begin{aligned} R'_\infty\left(u^\nu(x)\right)\delta_\infty^\nu &= a'\delta_\infty^\nu+\Delta t\nabla\cdot\left(\mathbf{f}'\delta_\infty^\nu\right)+\Delta t\nabla\cdot\left[\mathbf{G}'\nabla u^\nu(x)+\mathbf{G}\nabla\delta_\infty\right]+\Delta t w'\delta_\infty^\nu \\ &= \Delta t\mathbf{G}\nabla^2\delta_\infty^\nu+\Delta t\left(\mathbf{G}'\nabla u^\nu+\nabla\mathbf{G}+\mathbf{f}'\right)\cdot\nabla\delta_\infty^\nu \\ &\qquad +\left(a'+\Delta t\left(w'+\nabla.\mathbf{f}'+\nabla\cdot\left(\mathbf{G}'\nabla u^\nu\right)\right)\right)\delta_\infty^\nu \\ &= \Delta t\mathbf{G}\nabla^2\delta_\infty^\nu+\bar{\mathbf{q}}\left(x,u^\nu\right)\cdot\nabla\delta_\infty^\nu+\bar{p}\left(x,u^\nu\right)\delta_\infty^\nu. \end{aligned}$$

Similarly, applying the Fréchet operator on the boundary condition results in

$$\begin{cases} \Delta t\mathbf{G}\nabla^2\delta_\infty^\nu+\bar{\mathbf{q}}\left(x,u^\nu\right)\cdot\nabla\delta_\infty^\nu+\bar{p}\left(x,u^\nu\right)\delta_\infty^\nu+R_\infty\left(u^\nu\right) &= 0, \quad \text{in } \Omega, \\ \alpha_1\delta_\infty^\nu+\alpha_2\frac{\partial}{\partial\nu}\delta_\infty^\nu &= 0 \quad \text{in } \partial\Omega, \end{cases} \tag{3.1.3}$$

where

$$\bar{\mathbf{q}}\left(x,u^\nu\right):=\Delta t\left(\mathbf{G}'\nabla u^\nu+\nabla\mathbf{G}+\mathbf{f}'\right), \tag{3.1.4}$$

and,

$$\bar{p}\left(x,u^\nu\right):=a'+\Delta t\left(w'+\nabla.\mathbf{f}'+\nabla\cdot\left(\mathbf{G}'\nabla u^\nu\right)\right). \tag{3.1.5}$$

Note that in Equations 3.1.3 to 3.1.5, the divergence and gradient operators are with respect to space, whereas the prime notation is used to denote differentiation with respect to u. That is, for example,

$$a':=\frac{\partial a}{\partial u^\nu}\left(x,u^\nu\right),$$

denotes the derivative of the change in accumulation with respect to the state variable. In the case of a diagonal tensor coefficient for the viscous flux field, we also have that,

$$\mathbf{G}\nabla^2\delta_\infty^\nu := \sum_{i=1}^{d} G_{ii}\partial_i^2 u^\nu.$$

Equations 3.1.3 define the *infinite-dimensional Newton process*. Notice that by the linearity of the boundary value conditions in Equations 3.1.1 the resulting boundary conditions for the Newton problem are homogeneous, assuming that the initial guess satisfies the boundary conditions.

Introducing the iteration index $\nu = 0, \ldots$, the generated iterates are $u^\nu(x) \in W^* \subset E$. In this infinite form, Newton's method may be applied to solve Equation 3.1.1. Starting from an initial guess, $u^0 \in W^*$, the infinite-dimensional Newton updates,

$$\delta_\infty^\nu(x) := u^{\nu+1} - u^\nu, \quad \nu = 0, 1, \ldots,$$

are obtained by solving the linear Equation 3.1.3.

3.2 Prolongation of the finite problem onto an infinite problem

At each Newton iteration, $\nu = 0, 1, \ldots$, during the course of a numerical simulation, a corresponding discrete Newton iterate, U_h^ν, is available. Moreover, the corresponding discrete residual vector, $R_h(U_h^\nu)$, may be computed. In the proposed work, the discrete iterate and residual are first prolonged onto the continuous Banach spaces; i.e. $I_h^\infty U_h \in W^*$ and $I_h^\infty R_h(U_h) \in \hat{E}$ respectively. Then, at each Newton iteration the projections are used in the BVP operator that is defined by Equation 3.1.3. This leads to the sequence of infinite problems that are defined as,

$$\begin{cases} \Delta t\mathbf{G}(x, I_h^\infty U_h^\nu)\nabla^2\delta_\infty^\nu + \bar{\mathbf{q}}(x, I_h^\infty U_h^\nu)\cdot\nabla\delta_\infty^\nu + \bar{p}(x, I_h^\infty U_h^\nu)\delta_\infty^\nu + I_h^\infty R_h(U_h^\nu) & = 0, \quad \text{in } \Omega, \\ \alpha_1\delta_\infty^\nu + \alpha_2\frac{\partial}{\partial\nu}\delta_\infty^\nu & = 0 \quad \text{in } \partial\Omega, \end{cases} \tag{3.2.1}$$

where the coefficients $\bar{\mathbf{q}}$ and $\bar{p}$ retain their definitions in Equations 3.1.4 and 3.1.5 respectively. This prolongation is well-defined for bounded discretization triplets in the sense that was already defined.

The basic premise of the proposed methods rely on a connection between the discrete Newton update δ_h^ν and the restriction of the solution to Equation 3.2.1 onto the discrete mesh as shown in the previous chapter.

3.3 Analytical approximations for the support of Newton updates.

The infinite-dimensional Newton process for both flow and transport is defined by Equation 3.2.1 which is a linear, variable coefficient BVP on a general spatial domain. In general, analytical solutions to such problems are intractable. In particular, there are two aspects that make these problems challenging: the variable coefficients, and the geometry of the domain. In the previous chapter, it was shown that accurate solution to the linear infinite-dimensional Newton iteration is not sought. Instead an estimate is computed which encloses the support of the actual Newton update and hence provides a good estimate of the l_0-norm of the update vector. To visualize the concept of conservative estimate, we present a dummy example. For this example the function considered is the absolute value of

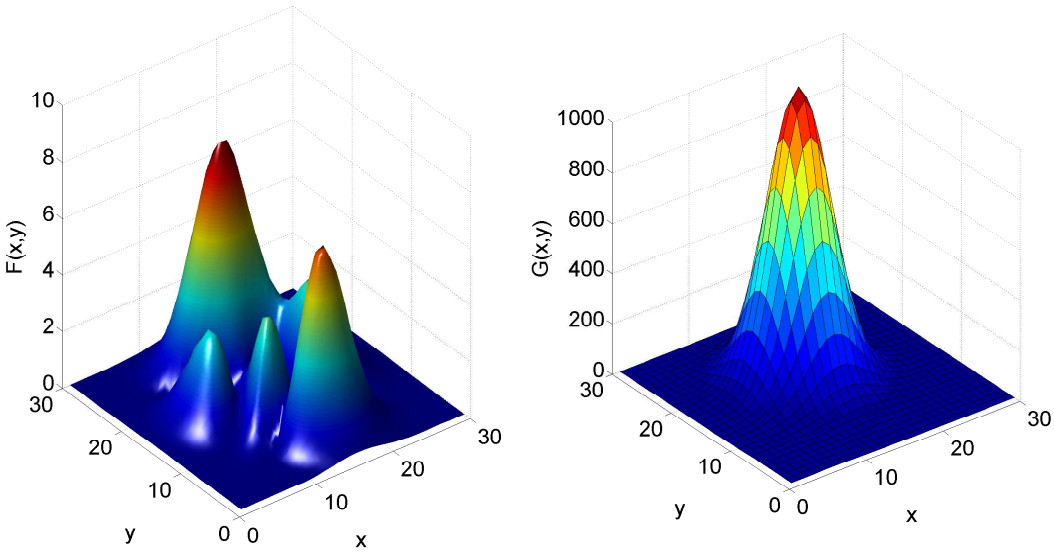
$$F(x,y) = 3(1-x)^2 e^{(-x^2-(y+1)^2)} - 10(\frac{x}{5} - x^3 - y^5)e^{(-x^2-y^2)} - \frac{1}{3}e^{(-(x+1)^2-y^2)},$$

where x and y are spatial coordinates. The left sub-figure of Figure 3.1a shows $F(x,y)$ for $x, y \in [-3,3]$. An estimate to the original function can be given as

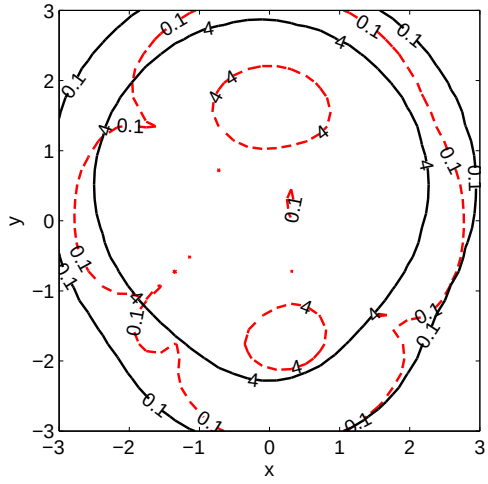
$$G(x,y) = 5(8.7-x)^2 e^{(-x^2-(y-0.5)^2)} - 50e^{(-x^2-(1+y)^2)}.$$

$G(x,y)$ is derived just for illustration purposes and there could be many estimates of this kind. The right sub-figure of Figure 3.1a shows a simple bump function, $G(x,y)$. Even though the values and the behavior of the two functions differ greatly, the cardinality (Figure 3.1b) of the estimate is conservative to that of the original function. The red (dashed)

contours refer to the original function while the black lines refer to the estimate derived. It is evident from the figure that $||F(x,y)||_0 \subset ||G(x,y)||_0$. In this context, $G(x,y)$, is called a *conservative estimate in the sense of support*. We will refer to this concept through out the development and implementation of the proposed mathematical framework and encourage the reader to return to this example in case of any confusion later in the text.



(a) The function and its estimate as a function of space variables.



(b) l_0 norm for F(x,y) and G(x,y) shown in Figure (a).

Figure 3.1: This figure shows the conservativeness of an estimate to the original function. Sub-figure(a) shows the function, F(x,y), and its estimate, G(x,y) as a function of two space dimensions. Sub-figure (b) shows the contours where the red (dashed) contours refer to the original function and the black contours refer to the estimate.

In this section we will devise model problems of the form in Equation 2.3.5 such that approximate solutions are readily available and are good approximations to the solutions of Equation 3.1.3 in the sense of support. The two key technical aspects towards achieving this are the homogenization of the variable coefficients, and an application of elliptic eigenvalue comparison theorems that will allow the use of surrogate spherical domains. The developments are specific to the various limiting cases that may be encountered in flow and transport problems but can be extended to any physical phenomena that fits the ADR form mentioned in the previous chapter.

Additionally, owing to the linearity of the projected infinite problem, we may apply superposition such that the source term in the BVP is a bump function with compact support. At each Newton iteration ν the corresponding discrete iterate U_h^ν is available, and the objective is to conservatively estimate the support of the Newton update δ_h^ν. By the linearity of the operator, superposition is applied by decomposing the residual vector as,

$$R_h\left(U_h^\nu\right) = \sum_{i=1}^{N} r_i \mathbf{e}_i$$

where r_i is the value of the residual vector along component i, and $\mathbf{e}_i$ is the i^{th} elementary unit vector. Hence, the Newton update may be obtained by summing the solutions of the sequence of subproblems,

$$R'_\infty\left(I_h^\infty U_h^\nu\right)\delta_\infty^{\nu,i} + r_i \mathbf{e_i} = 0 \qquad i = 1, \ldots, N. \tag{3.3.1}$$

This also implies that the support of the Newton update is contained in the union of the support sets of the subsolutions,

$$\operatorname{supp} \delta_h^\nu \subseteq \bigcup_{i=1}^{N} \operatorname{supp} \delta_\infty^{\nu,i}.$$

Closed form analytical solutions to Equation 3.3.1 are seldom tractable. As mentioned before the two assumptions to reduce this equation to an easier solvable equation are specific to

the type of problem being solved. The homogenization and solution techniques that are applicable to these two limiting cases (elliptic and hyperbolic) are rather distinct. In this section, we develop solutions to both cases, allowing for general two-phase flow as well as the commonly applied limiting cases.

3.4 The general elliptic form.

In the case of flow, the independent variable is pressure, and subsequently the inviscid flux term incorporates the effects of gravity and capillarity. The viscous flux is attributed to the multiphase extension to Darcy's law. Except in the specific cases of negligible capillary and buoyancy effects, none of the coefficients in Equations 3.2.1 and 3.1.1 are uniformly zero throughout the domain. The elliptic form is characterized by a viscous coefficient term $\mathbf{G}$ that is not identically equal to zero throughout the domain. We develop the solutions assuming an isotropic field where $\mathbf{G} = k\mathbf{1}$. The diagonal anisotropic case can be obtained in a similar manner by application of a standard change in the spatial coordinates.

Given a Newton iterate $u := I_h^\infty U_h$ for pressure or saturation, the infinite-dimensional Newton update δ_∞ can be obtained by solving the linear BVP:

$$\begin{cases} \Delta\delta_\infty + \mathbf{q}(x,u)\cdot\nabla\delta_\infty + p(x,u)\,\delta_\infty + I_h^\infty R_h/\Delta t k & = 0, \quad \text{in } \Omega, \\ \frac{\partial}{\partial\nu}\delta_\infty & = 0, \quad \text{in } \partial\Omega, \end{cases} \tag{3.4.1}$$

where,

$$\begin{aligned} \mathbf{q}(x,u) &:= \frac{1}{k}\left(k'\nabla u + \nabla_x k + \mathbf{f}'\right) \\ &= \frac{1}{k}\left(\nabla k + \mathbf{f}'\right) \end{aligned}$$

and,

$$p(x,u) := \frac{a'}{\Delta t k} + \frac{w' + \nabla.\mathbf{f}' + \nabla\cdot(k'\nabla u)}{k},$$

and k is the isotropic viscous flux coefficient, and k' is its derivative with respect to the state

variable.

As demonstrated in Appendix C, it is possible to introduce a variable transformation that recasts Equations 3.4.1 into the linear screened Poisson equation with a spatially variable screening parameter. In particular, let

$$\alpha := \frac{1}{2} \int_{\zeta[x_0,x]} \mathbf{q}\left(x,u\right) \cdot dx,$$

where ζ is any curve in Ω emanating from a point $x_0 \in \Omega$. This definition requires that $\mathbf{q}$ be line-integrable bounded. Next, we can introduce the transformation,

$$\hat{\delta}_p := \exp\left(\alpha\left(x,u\right)\right) \delta_\infty.$$

Then δ_∞ is a solution to Equation 3.4.1 if and only if $\hat{\delta}_p$ is a solution to the screened Poisson equation,

$$\begin{cases} \Delta\hat{\delta}_p - \lambda^2\left(x,u\right)\hat{\delta}_p + g = 0, & x \in \Omega \\ \left(\nabla\hat{\delta}_p - 0.5\hat{\delta}_p\mathbf{q}\left(x,u\right)\right) \cdot \hat{n} = 0 & x \in \partial\Omega \end{cases}, \tag{3.4.2}$$

where

$$\lambda^2\left(x,u\right) := \frac{1}{2}\nabla \cdot \mathbf{q} + \mathbf{q} \cdot \mathbf{q} - p\left(x,u\right), \tag{3.4.3}$$

and

$$g := \exp\left(\alpha\left(x,u\right)\right) I_h^\infty R_h/\Delta t k.$$

While this well-studied equation is linear, the variable screening parameter and the general geometry of the domain prohibit a closed form analytical solution. So far, no approximations have been made, and $\hat{\delta}_p$ is an accurate analytical approximation to the Newton update up to the error introduced by the discretization scheme. Next, we introduce two approximations that will lead to simple closed form estimates to $\hat{\delta}_p$.

3.4.1 Approximation 1: homogenization

Several homogenization strategies may be applied in order to derive a constant coef-

ficient form of Equation 3.4.2. That is, we will introduce a constant parameter $\lambda^* \in \mathbb{R}$, such that the analytical solution δ_p^* to the constant coefficient problem,

$$\begin{cases} \Delta\delta_p^* - \lambda^{*2}\delta_p^* + g = 0, & x \in \Omega \\ \left(\nabla\delta_p^* - 0.5\delta_p^*\mathbf{q}^*\right)\cdot\hat{n} = 0 & x \in \partial\Omega \end{cases}, \tag{3.4.4}$$

is in some sense a good approximation to $\hat{\delta}_p$. Two particular homogenization strategies are considered. The first is the result of a standard two-scale asymptotic expansion (see for example [26]), and leads to the volume averaged screening parameter,

$$\lambda_1^* := \frac{\int_\Omega \lambda\left(x,u\right)dx}{\int_\Omega dx}. \tag{3.4.5}$$

This choice is analogous to the harmonic average for the homogenization of permeability. Solving Equation 3.4.4 with Condition 3.4.5 however, is *not* guaranteed to produce estimates that are conservative in the sense of support. However, the estimates will preserve continuity as does the harmonic averaging for diffusion operators with variable permeability. On the other hand, if we homogenize as,

$$\lambda_2^* := \inf_\Omega \lambda\left(x,u\right), \tag{3.4.6}$$

then the corresponding solution to Equation 3.4.4 is a conservative estimate to $\hat{\delta}_p$ in the sense of support. This is easily demonstrated using the theorem below. These estimates may be overly conservative however.

Theorem 3. *Given a linear second order variable coefficient Equation 3.4.2 on Ω, conservative estimates, $||\hat{\delta}_p(x)||_0 \subseteq ||\delta_p^*(x)||_0$, can be derived by solving a constant coefficient equation of the form*

$$\begin{cases} \Delta\delta_p^* - \lambda^{*2}\delta_p^* + g = 0, & x \in \Omega \\ \left(\nabla\delta_p^* - 0.5\delta_p^*\mathbf{q}^*\right)\cdot\hat{n} = 0 & x \in \partial\Omega \end{cases}, \tag{3.4.7}$$

such that the constant parameter $\lambda^* = \inf_\Omega \lambda(x)$.

Proof. Subtracting Equations 3.4.7 and 3.4.2 we get

$$\begin{cases} \Delta\delta_p^* - \Delta\hat{\delta}_p - \lambda^{*2}\delta_p^* + \lambda^2(x,u)\,\hat{\delta}_p = 0, & x \in \Omega \\ \left(\nabla\delta_p^* - 0.5\delta_p^*\mathbf{q}^* - \nabla\hat{\delta}_p + 0.5\hat{\delta}_p\mathbf{q}\right)\cdot\hat{n} = 0 & x \in \partial\Omega \end{cases}.$$

Introducing a variable $\bar{\delta} = \delta_p^* - \hat{\delta}_p$, and further reducing the above equation, we obtain

$$\begin{cases} \Delta\bar{\delta} - \lambda^{*2}\bar{\delta} = -(\lambda^2(x,u) - \lambda^{*2})\hat{\delta}_p, & x \in \Omega \\ \left(\nabla\bar{\delta} - 0.5\bar{\delta}\mathbf{q}^*\right)\cdot\hat{n} = -\left(0.5\hat{\delta}_p(\mathbf{q}-\mathbf{q}^*)\right)\cdot\hat{n} & x \in \partial\Omega \end{cases}. \tag{3.4.8}$$

Equation 3.4.8 is a constant coefficient modified Helmholtz equation with a variable source term. A strong conclusion holds for a second order elliptic equation such as the one stated above, namely a *strong maximum principle* [11], when $-\lambda^* < \lambda_1$, where λ_1 represents the principle eigenvalue of $-\Delta$. If the right-hand side of Equation 3.4.8 is nonnegative and not identically zero in the domain, then the solution is negative in the domain, given certain assumptions on the boundary condition. We refer to the classical book [43] for more details. [9] and [44] give additional details on this important tool. For this system of equations, the sign of the solution, $\bar{\delta}$, will depend on $(\lambda^2(x,u) - \lambda^{*2})$ and $\hat{\delta}_p$. To ensure a conservative estimate for a positive $\hat{\delta}_p$, the solution of Equation 3.4.8, $\bar{\delta}$, has to be positive, $\left(\nabla\bar{\delta} - 0.5\bar{\delta}\mathbf{q}^*\right)\cdot\hat{n} \geq 0$ for $x \in \partial\Omega$ and $\mathbf{q}^* < 2\lambda^*$. The reasoning behind the condition on $\mathbf{q}^*$ is given in the next theorem. On the other hand, if $\hat{\delta}_p$ is negative, $\bar{\delta}$ has to be negative. For a positive value of the residual function in the domain, the value of $\bar{\delta}$ has to be positive. Subsequently, to ensure conservative estimate in either of the cases

$$(\lambda^2(x,u) - \lambda^{*2}) \geq 0.$$

The above condition is satisfied for $\lambda^* = \inf_\Omega \lambda(x,u)$. Therefore, taking the infimum of the

variable coefficient, $\lambda(x, u)$ results in

$$\left|\delta_p^*\right| \geq \left|\hat{\delta}_p\right|.$$

□

The choice between the two strategies provides alternative trade-offs between aggressive localization and the risk of missing some non-negligible entries in the Newton update.

Example 2. *Given a variable coefficient second order equation of the form*

$$y''(x) - xy(x) = -x^2, \quad x \in [0, 1], \tag{3.4.9}$$

show that conservative solutions can be obtained by solving a constant coefficient equation using Condition 3.4.6, for

(i) $y'(0) = 0$ *and* $y'(1) = 0$

(ii) $y(0) = 0$ *and* $y(1) = 0$

(iii) $y'(0) + by(0) = 0$ *and* $y'(1) - by(1) = 0$, *given* b = *-2.*

Solution. The analytical solution of Equation 3.4.9 is given by

$$y(x) = c_1 Ai(x) + c_2 Bi(x) + \pi x Ai(x) Bi'(x) - \pi x Ai'(x) Bi(x), \tag{3.4.10}$$

where Ai is the Airy function of the first kind, Bi is the Airy function of the second kind, the prime notations are the derivative of the functions and c_1 and c_2 are constants that will be determined from the three boundary conditions. The infimum of the variable coefficient, shown in Condition 3.4.6, in this case is zero. The constant coefficient, for Equation 3.4.7, is chosen to be 0.0001. The solutions for the three cases are shown in Figure 3.2. Figure 3.2a shows the results for the Dirichlet boundary condition as given in (i), while Figures 3.2b and 3.2c shows the results for the Neumann, (ii) and the Robin, (iii), problems, respectively.

In the case of the Robin boundary condition, b is a negative number because as stated in the proof above, $q^* < 2\lambda^*$ and in this case, $\lambda^* = 0$. Hence, the values of q^* for which the estimate will be conservative will be negative.

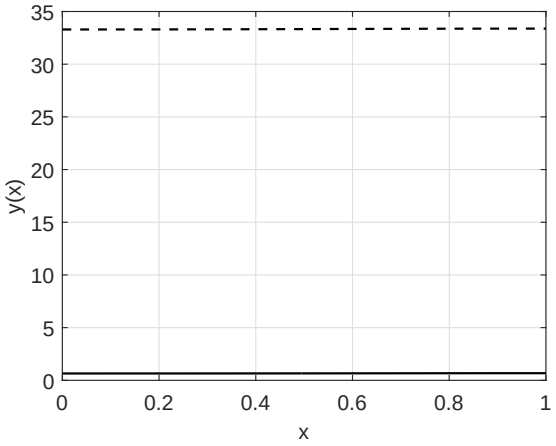
In each of the cases, dashed lines are obtained by solving the homogenized equation using Condition 3.4.6 and the solid line is obtained by plotting Equation 3.4.10 for $x \in [0, 1]$. In each of the cases, homogenized estimate results in a conservative estimate in the sense of the absolute values and support.

Example 3. *Show the conservativeness of the estimate obtained by solving the constant coefficient equation for the same problem as shown in Example 1 with the boundary conditions given in (iii), with Condition 3.4.5 instead of the infimum strategy.*

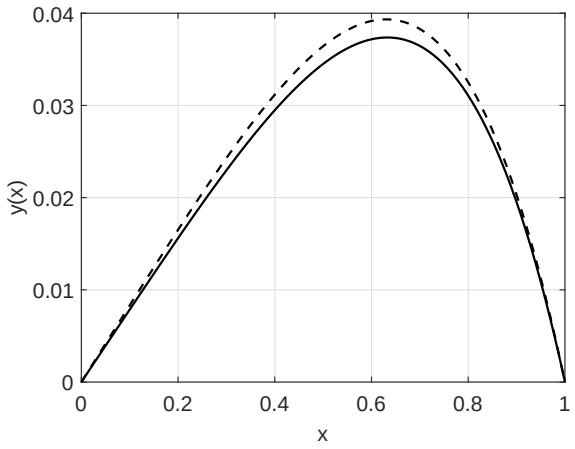
Solution. As mentioned before, using Condition 3.4.5 to obtain a homogenized equation might not result in conservative estimate to Equation 3.4.2 in the sense of the magnitude of the solution. Figure 3.3 shows the comparison of the two solutions for Equation 3.4.9. Dashed line shows the homogenized estimate while the solid line is obtained by evaluating Equation 3.4.10 for $x \in [2, 3]$. As expected the result is not conservative in the magnitude. An important observation in this example is that for the values of the screening parameter greater than the homogenized value, Condition 3.4.5 results in conservative estimates. Only for the coefficient values less than the homogenized coefficient result in a nonconservative estimate. Also, the solution obtained is quite close to the original solution as opposed to the infimum strategy and hence Condition 3.4.5 might be the preferred homogenization strategy for cases with small variance in the variable coefficient values.

3.4.2 Approximation 2: spherical symmetry

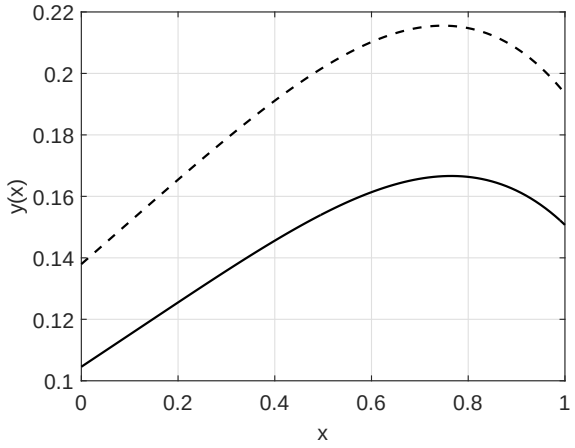
Equation 3.4.7 resembles the eigenvalue problem of the Laplacian with the exception that it is inhomogeneous. A well-known property of the Laplacian's eignevalue problems are the comparison theorems that lead to various forms of maximum principles. One particular example of such a result is that of the ordering of eigenvalues corresponding to domains that are successively contained within each other [20]. The second approximation we will make



(a) Dirichlet boundary condition



(b) Neumann boundary condition



(c) Robin boundary condition

Figure 3.2: This figure shows the conservative estimates obtained by solving a homogenized second order equation, dashed line, with the screening parameter equal to the infimum of the variable screening parameter in Equation 3.4.9, shown by the bold line.

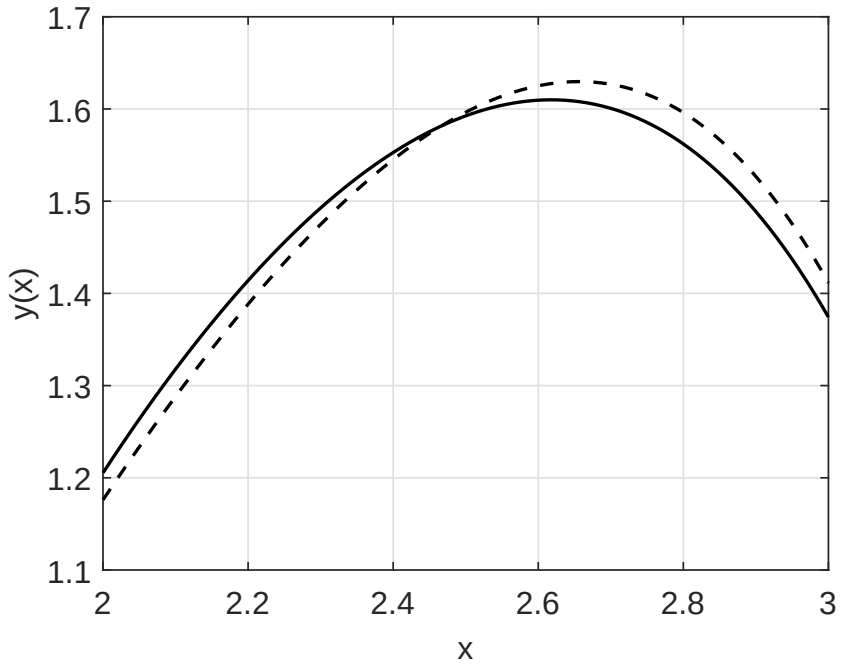


Figure 3.3: This figure shows the nonconservative estimate obtained by solving a homogenized second order equation, dashed line, with the screening parameter satisfying Condition 3.4.5 for Equation 3.4.9, shown by the bold line.

is to consider the specific spherical domain $\Omega_2 = \{x : \|x\|_2 \leq R\}$ of radius R that shares the same centroid as Ω.

Theorem 4. *Given a constant coefficient second order ODE defined on Ω_1, as shown in Figure 3.4, such that*

$$\begin{cases} \Delta\delta_p^* - \lambda^{*2}\delta_p^* + g = 0, & x \in \Omega_1 \\ \left(\nabla\delta_p^* - 0.5\delta_p^*\mathbf{q}^*\right)\cdot\hat{n} = 0 & x \in \partial\Omega_1 \end{cases}, \tag{3.4.11}$$

a conservative estimate can be obtained by solving a constant coefficient equation with identical screening parameter, given by

$$\begin{cases} \Delta\delta^* - \lambda^{*2}\delta^* + g = 0, & x \in \Omega_2 \\ (\nabla\delta^* - 0.5\delta^*\mathbf{q}^*)\cdot\hat{n} = 0 & x \in \partial\Omega_2 \end{cases}, \tag{3.4.12}$$

for $\Omega_1 \subseteq \Omega_2$.

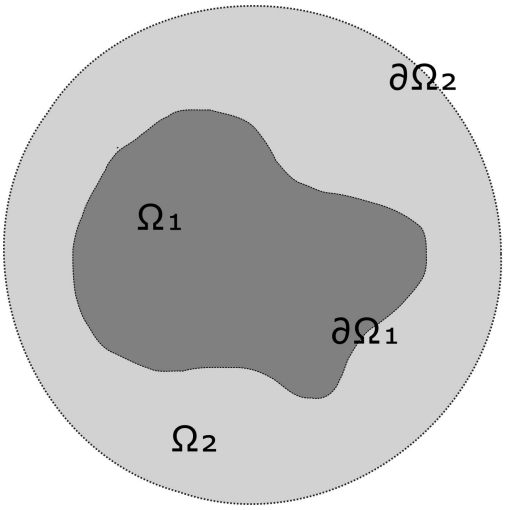


Figure 3.4: Figure for Theorem 4.

Proof. Let $\bar{\delta} = \delta^* - \delta_p^*$ and $(\nabla\delta^* - 0.5\delta^*\mathbf{q}^*)\cdot\hat{n} = \phi(x)\ \forall x \in \partial\Omega_1$. Subtracting Equations 3.4.12 and 3.4.11 and evaluating on Ω_1, we get

$$\begin{cases} \Delta\bar{\delta} - \lambda^{*2}\bar{\delta} = 0, & x \in \Omega_1 \\ \left(\nabla\bar{\delta} - 0.5\bar{\delta}\mathbf{q}^*\right)\cdot\hat{n} = \phi(x) & x \in \partial\Omega_1 \end{cases}. \tag{3.4.13}$$

In Equation 3.4.13, $\phi(x)$ is the value of $(\nabla\delta^* - 0.5\delta^*\mathbf{q}^*)\cdot\hat{n}$ evaluated on $\partial\Omega_1$. If we consider the case of δ^* to be positive, we know from Theorem 3 that $\delta^* \geq 0$ on $\partial\Omega_1$ boundary as well. This implies the property of monotonicity for the monotone Helmholtz equation. This directly results in $\phi(x) \geq 0$. Let $\mathcal{L} = \Delta - \lambda^{*2}$, we know that

$$-\mathcal{L}\bar{\delta} \geq 0 \ \ \forall x \in \Omega_1,$$

and given the conditions on δ^* in $\partial\Omega_1$ and $\mathbf{q}^* < 2\lambda^*$ in case $\mathbf{q}^*$ is positive or for any $\mathbf{q}^* < 0$

$$\bar{\delta} \geq 0 \ \ \forall x \in \partial\Omega_1.$$

Hence, from the positivity preserving property, we can conclude that either $\bar{\delta}(x) > 0$ for $x \in \Omega_1$ or $\bar{\delta} \equiv 0$. Either way, we have a conservative estimate for the solution of a constant

coefficient second order equation on an irregular domain. Due to the positive value of λ^*, in cases where q^* is negative, $\bar{\delta}$ is always unconditionally positive and hence the solution is always conservative. The relation between q^* and λ^* is obtained in the example shown. □

Example 4. *Given $\phi(x) = 1.0$ and $y(x) = \bar{\delta}(x)$, Equation 3.4.13 in one dimension can be rewritten as*

$$\begin{cases} y''(x) - \lambda^{*2} y(x) = 0, & x \in [0,1] \\ y'(0) + 0.5y(0)q^* = -1.0 & \\ y'(1) - 0.5y(1)q^* = 1.0 & \end{cases}, \tag{3.4.14}$$

with boundaries at $x = 0$ and $x = 1$. Show that $y(x) > 0$ for $x \in \partial\Omega$, given $q^* < 2\lambda^*$.

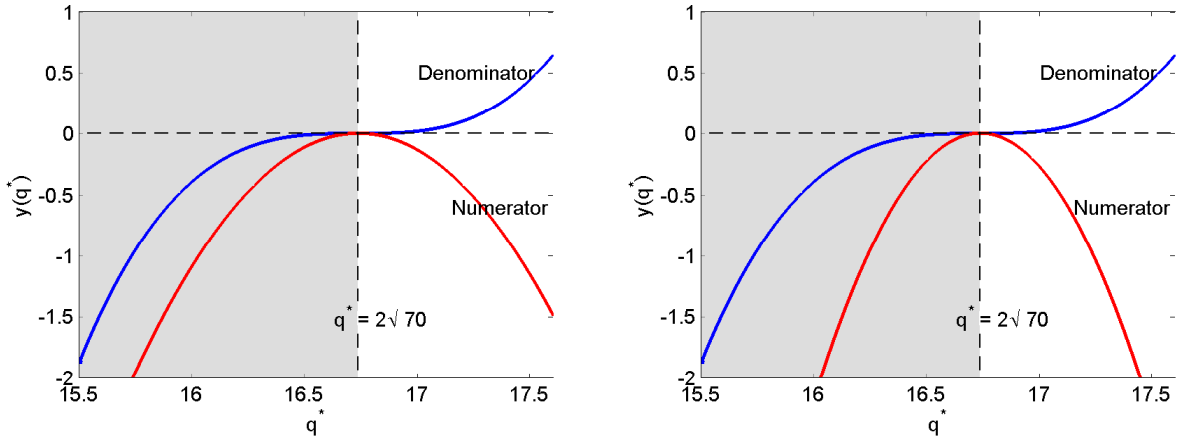
Solution. The solution of Equation 3.4.14 can be written as

$$\begin{aligned} y(x) = \Big[e^{-\lambda^* x} \big(\left(\lambda^{*2} - 0.25q^{*2}\right) e^{2\lambda^* x} + \left(-\lambda^* q^* + \lambda^{*2} + 0.25q^{*2}\right) e^{2\lambda^*(x+0.5)} \\ + e^{\lambda^*} \left(\lambda^{*2} - 0.25q^{*2}\right) + e^{2\lambda^*} \left(-\lambda^* q^* + \lambda^{*2} + 0.25q^{*2}\right) \big) \Big] \\ \Big/ \Big[\left(\lambda^* - 0.5q^*\right) \left(e^{2\lambda^*} \left(-\lambda^* q^* + \lambda^{*2} + 0.25q^{*2}\right) - \lambda^* q^* - \lambda^{*2} - 0.25q^{*2}\right) \Big] \end{aligned} \tag{3.4.15}$$

With further analysis it can be seen that the numerator is negative with varying x for most part and the sign of the denominator depends on the $\lambda^* - 0.5q^*$ term. The second term in the denominator is always negative. Hence $y(x)$ will be positive when $\lambda^* - 0.5q^* > 0$ which is at $q^* < 2\lambda^*$. To test this claim, assuming $\lambda^{*2} = 70$, the above equation results in

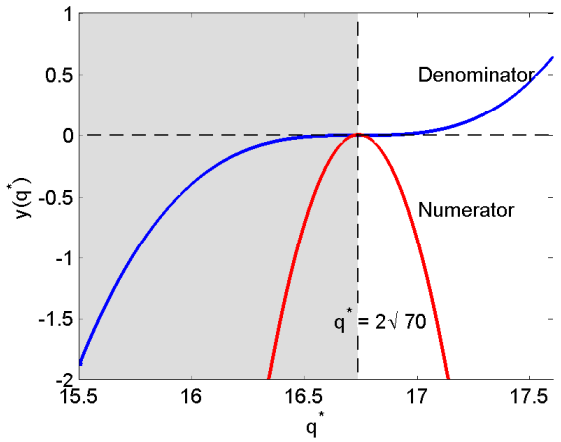
$$y(x) = \frac{e^{-\sqrt{70}x} \left(e^{2\sqrt{70}x} (q^*(0.01 \ - 0.0004q^*) - 0.13) + q^*(66.93 \ - 1.99q^*) - 560.13 \right)}{(q^{*3} - 50.19q^{*2} + 840q^* - 4685.30)} \tag{3.4.16}$$

Figure 3.5 shows the plots of $y(x)$ with changing q^* for four different cases of x. In all the four sub-figures, the gray shaded area marks values of q^* that results in $y(x) \geq 0$. The vertical line that divides the two areas turns out to be at $q^* = 2\sqrt{70} = 16.73$. Figure 3.6 shows the solution of Equation 3.4.14 for $\lambda^* = 70$ and varying values of $q^* < 16.73$. All the

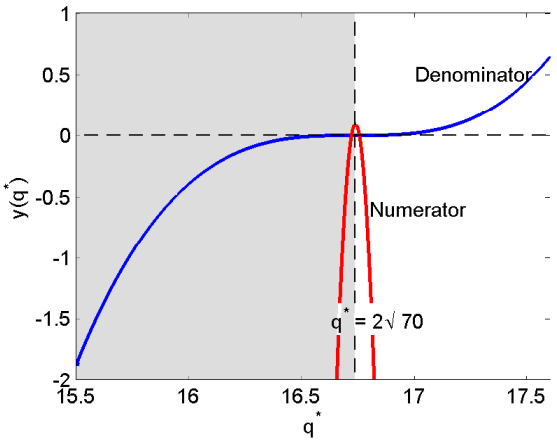


(a) $x = 0.1$

(b) $x = 0.5$



(c) $x = 0.6$



(d) $x = 0.8$

Figure 3.5: This figure shows the behavior of $y(x)$ given by Equation 3.4.16 in terms of its numerator and denominator. For different values of x, the numerator, represented by the red line, always result in negative values while only the concavity changes. The sign of the denominator changes with the value of q^* and it can be easily observed that the inflection point lies at $q^* = 2\sqrt{70}$, thus proving our previous deduction.

solutions are positive and hence confirm the conservative behavior of Equation 3.4.12. On the other hand, Figure 3.7 shows the solution for $q^* > 16.73$, which result in negative values thus confirming the nonconservative behavior of Equation 3.4.12.

3.4.3 Solutions for the estimates.

Owing to the constant screening parameter and the spherical geometry of the domain, the solution to Equation 3.4.12 will be radially symmetric. The symmetric form of

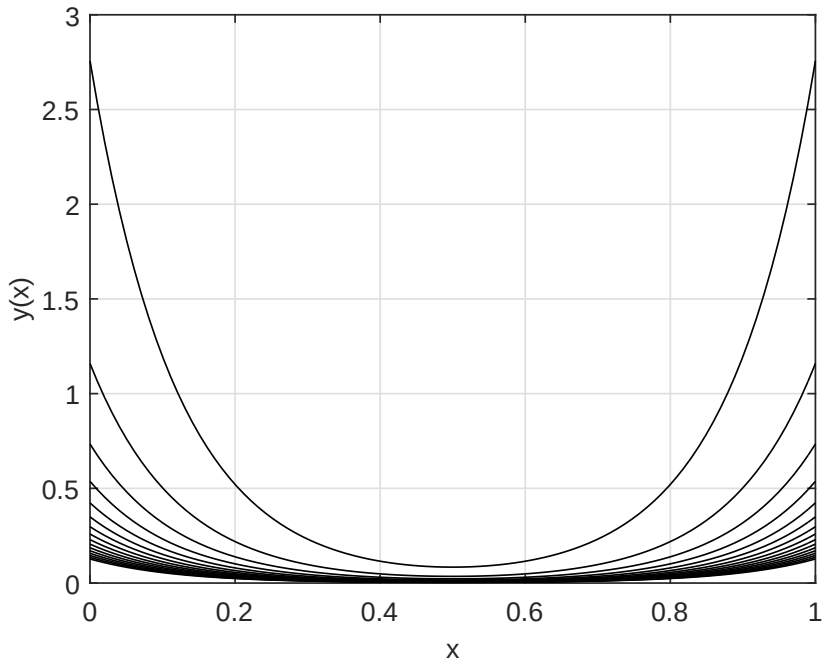


Figure 3.6: Figure shows $y(x) > 0$ for varying values of $q^* < 2\lambda^*$ and $\phi(x) > 0$ for Example 4.

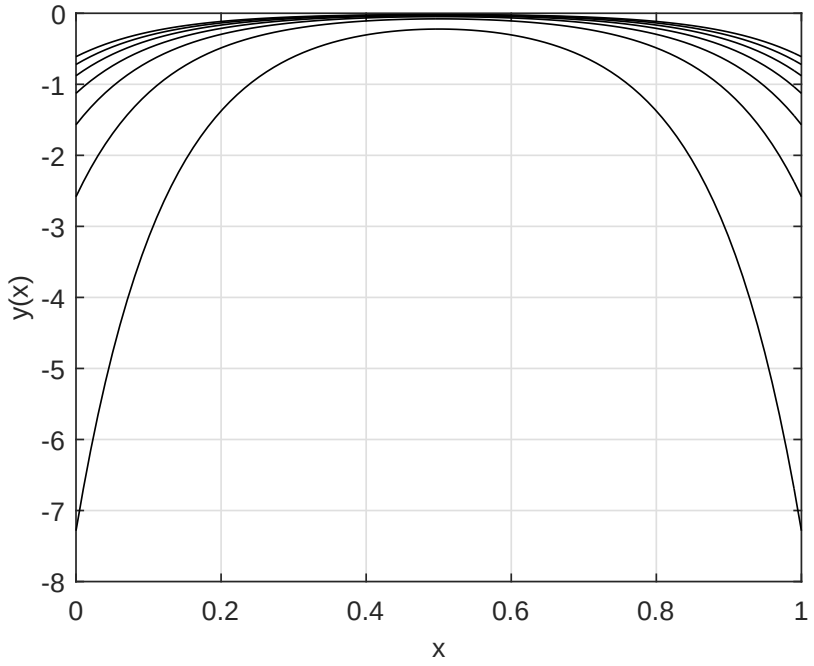


Figure 3.7: Non-positive behavior of $y(x)$ for varying values of $q^* > 2\lambda^*$ and $\phi(x) > 0$ for Example 4.

Equation 3.4.12 is,

$$\begin{cases} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\delta^*(r)}{\partial r}\right) - \lambda^{*2}\delta^*(r) = g, & r \leq R \\ \left(\frac{\partial\delta^*}{\partial r} - 0.5\delta^*\mathbf{q}^*\right) = 0 & r = R \end{cases}, \tag{3.4.17}$$

The analytical solution to the Neumann problem is readily available using a boundary integral solution method with the fundamental solution [20]. That is, the solution can be written in terms of an integral equation given by

$$\begin{aligned} \delta^*(r) &= \int_\Omega G(r,r^*)\Delta\delta^*(r^*)d\Omega - \int_{\partial\Omega}\left[G(r,r^*)\frac{d\delta^*(r^*)}{d\eta_{r^*}} - \delta^*(r^*)\frac{dG(r,r^*)}{d\eta_{r^*}}\right]d\partial\Omega \\ &= \int_\Omega G(r,r^*)\Delta\delta^*(r^*)d\Omega + \int_{\partial\Omega}\delta^*(r^*)\frac{dG(r,r^*)}{d\eta_{r^*}}d\partial\Omega \\ &= -\int_\Omega G(r,r^*)g(r^*)d\Omega, \end{aligned}$$

where $r, r^* \in \Omega$, η_{r^*} is the unit outward normal and $G(r,r^*)$ is the Green's function given by

$$G(r,r^*) = \Psi(||r-r^*||) + \tilde{h}(r^*). \tag{3.4.18}$$

In the above equation, $\Psi(||r-r^*||)$ is the fundamental solution of the modified Helmholtz equation and is given by,

$$\Psi(||r-r^*||) = \frac{-1}{2\lambda}e^{(-\lambda^*|r-r^*|)}$$

and $\tilde{h}(r^*)$ is called the corrector function which is the solution of,

$$\begin{cases} \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\tilde{h}^*(r)}{\partial r}\right) - \lambda^{*2}\tilde{h}^*(r) & = 0 \quad 0 \leq r < R \\ \frac{d\tilde{h}}{dr^*} - \frac{d\Psi(||r-r^*||)}{dr^*} & = 0 \quad r = R \end{cases}. \tag{3.4.19}$$

The solution is obtained by evaluating the corrector function, $\tilde{h}(r^*)$, and substituting the expression in Equation 3.4.18. Finally, the solution for a point residual source is derived by integrating the Green's function with the inhomogeneous term over the domain. The

solution is given by,

$$\delta^*(r) = \frac{g^{l^*}\,\Delta r}{\Delta t k(p^\nu, r)}\left[\frac{\mathbf{I}_0(\lambda^* l^*)}{2\lambda^* \mathbf{I}_1(\lambda^* R)} exp[-\lambda^*(R-r)] - \frac{1}{2\lambda^*} exp[-\lambda^*|r-l^*|]\right] \quad r, l^* \in \Omega, \tag{3.4.20}$$

where g^{l^*} is the nonzero residual term at the spatial coordinate l^*, $\mathbf{I}_0$ and $\mathbf{I}_1$ are modified Bessel's functions of the first kind and Δr is the thickness of the grid block in radial coordinates.

For a non-trivial residual source, each bump function has a nonzero support. Assuming that the support extends from 0 to l_1, where 0 represents the origin shifted onto the center of the grid cell containing the nonzero residual, we can obtain the solution

$$\delta^*(r) = -\int_0^{l_1} G(r, r^*) g(r^*) d\Omega$$

$$\delta^*(r) = \frac{g^{l^*}}{\Delta t k(p^\nu, r)}\left[\frac{l_1\left(\mathbf{I}_0(\lambda^* l_1)(\pi \mathbf{L}_1(\lambda^* l_1) + 2) - \pi \mathbf{I}_1(\lambda^* l_1)\mathbf{L}_0(\lambda^* l_1)\right)}{4\lambda^* \mathbf{I}_1(\lambda^* R)} exp[-\lambda^*(R-r)] - \frac{1}{2\lambda^{*2}} exp[-\lambda^*(r-l_1)]\right] \quad r, l_1 \in \Omega. \tag{3.4.21}$$

In the above equation, l^* is the position of the source term that vanishes when the origin is centered at the nonzero residual source term. R is the reservoir boundary, g^{l^*} is the component of the residual function evaluated in the gridcell containing l^* and $\mathbf{L}_0$ and $\mathbf{L}_1$ are modified Struve functions.

3.4.4 The radius of the nonzero support set.

An alternative to computing the estimate over the mesh is to compute the radius away from the nonzero residual at which point the update estimate decays monotonically below a certain threshold. Each nonzero residual term causes a strictly decaying update in a radial fashion. This allows us to characterize the radius of the nonzero support caused by a single nonzero residual value. In this algorithm, instead of solving for the estimates of the

Newton update for every grid cell corresponding to each nonzero source term, we find the radius of influence for each nonzero source term, such that the decaying Newton's update has reached a cut-off value. In this work, we use ϵ_p which is by default the machine precision number. The radius of influence within which grid cells will be flagged is the solution to,

$$\epsilon_p = \frac{g^{l^*}}{\Delta tk(p^\nu, r)} \Bigg[\frac{l_1 \left(\mathbf{I}_0(\lambda^* l_1)(\pi \mathbf{L}_1(\lambda^* l_1) + 2) - \pi \mathbf{I}_1(\lambda^* l_1)\mathbf{L}_0(\lambda^* l_1)\right)}{4\lambda^* \mathbf{I}_1(\lambda^* R)} exp[-\lambda^*(R - r)] \\ - \frac{1}{2{\lambda^*}^2} exp[-\lambda^*(r - l_1)] \Bigg] \tag{3.4.22}$$

$$\epsilon_p \frac{\Delta tk(p^\nu, r)}{g^{l^*}} exp[\lambda^* r] = \Bigg[\frac{l_1 \left(\mathbf{I}_0(\lambda^* l_1)(\pi \mathbf{L}_1(\lambda^* l_1) + 2) - \pi \mathbf{I}_1(\lambda^* l_1)\mathbf{L}_0(\lambda^* l_1)\right)}{4\lambda^* \mathbf{I}_1(\lambda^* R)} exp[-\lambda^* R]\ exp[2\lambda^* r] \\ - \frac{1}{2{\lambda^*}^2} exp[\lambda^* l_1] \Bigg]$$

The above equation can be recast into a quadratic equation in $exp(\lambda^* r)$. Subsequently, the solution can be obtained by solving for the roots of the quadratic equation and then finding the radius by taking the logarithm of the solution. The radius is hence give by,

$$r = \ln \left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right) / \lambda^* \tag{3.4.23}$$

where

$$A = \frac{l_1 \left(\mathbf{I}_0(\lambda^* l_1)(\pi \mathbf{L}_1(\lambda^* l_1) + 2) - \pi \mathbf{I}_1(\lambda^* l_1)\mathbf{L}_0(\lambda^* l_1)\right)}{4\lambda^* \mathbf{I}_1(\lambda^* R)} exp[-\lambda^* R] \tag{3.4.24}$$

$$B = -\epsilon_p \frac{\Delta tk(p^\nu, r)}{g^{l^*}} \tag{3.4.25}$$

$$C = -\frac{1}{2{\lambda^*}^2} exp[\lambda^* l_1] \tag{3.4.26}$$

It is necessary to determine a physical value of the radius from the two roots of the quadratic equation. Once we have the radius, the cells contained within the circle centered at the source

location with radius determined by the roots, are flagged. The union of all of the flagged cells will give us the total size of the domain which will experience a nonzero Newton update. The expression inside the logarithm in Equation 3.4.23 will be greater than 1. If the radius evaluates to zero, physically speaking there will be no pressure update propagation.

Following the same procedure, we can obtain the estimates in three dimensions. The estimates are obtained by solving the pressure equation in spherical coordinates. The final solution is given by

$$\delta^*(r) = \frac{g^{l^*} exp[\lambda^*(l_1 - r)]}{2\lambda^* \Delta t k(p^\nu, r)} \left[\frac{R^2}{r(1+\lambda^* R)} - \frac{1}{\lambda^*} \right] \tag{3.4.27}$$

3.5 The hyperbolic limiting case.

For the transport problem, the independent variable may be saturation. In this case, the inviscid flux is the Darcy flow, whereas the viscous flux is solely due to capillarity. Under the assumption of negligible capillarity, the infinite-dimensional Newton iteration simplifies to the hyperbolic form

$$\begin{cases} \mathbf{q}\left(x, u^\nu\right) \cdot \nabla \delta_\infty^\nu + p\left(x, u^\nu\right) \delta_\infty^\nu + I_h^\infty R_h & = 0, \quad \text{in } \Omega, \\ \delta_\infty^\nu & = 0, \quad \text{in } \partial\Omega, \end{cases} \tag{3.5.1}$$

where now,

$$\mathbf{q}\left(x, u^\nu\right) := \Delta t \mathbf{f}',$$

and,

$$p\left(x, u^\nu\right) := a' + \Delta t \left(w' + \nabla . \mathbf{f}'\right),$$

since $\mathbf{G} \equiv 0$. While this well-studied equation is linear, the variable screening parameter and the general geometry of the domain prohibit a closed form analytical solution. So far, no approximations have been made, and δ_∞^ν is an accurate analytical approximation to the Newton update up to the error introduced by the discretization scheme. Next, we introduce two approximations that will lead to simple closed form estimates to δ_∞^ν. From this point

forward, we will drop the iteration index, ν.

3.5.1 Approximation 1: homogenization.

While $\mathbf{q}$ is spatially variable over Ω, we will consider a homogenized form that guarantees conservative estimates in the sense of support. Similar to the approximations developed for the case of flow, the first step is to homogenize the variable coefficients and then to project the equation onto a regular domain enclosing the original domain. In usual cases, the enclosing domain is assumed to be either radial or spherical, depending on the dimensions of the problem. This allows us to derive radially or spherically symmetric solutions and compute the radius of effect as shown in the previous section. The following proofs show the theoretical justification behind these approximations for first order partial differential equations.

Theorem 5. *For a given first order variable coefficient differential equation of the form*

$$q_x(x,y,u)\frac{\partial \delta_\infty}{\partial x} + q_y(x,y,u)\frac{\partial \delta_\infty}{\partial y} + p(x,y,u)\delta_\infty + I_h^\infty R_h = 0, \tag{3.5.2}$$

where $x \in \Omega \subset \mathbb{R}^2$, conservative estimate in the sense of support can be obtained by solving an equation of the form

$$\frac{\partial \delta^*}{\partial y} + q^*\frac{\partial \delta^*}{\partial x} + \gamma^*\delta^* + \frac{I_h^\infty R_h}{q_y(x,y,u)} = 0, \tag{3.5.3}$$

where $q^* := \max(q_x(x,y,u)/q_y(x,y,u))$ and $\gamma^* := \min(p(x,y,u)/q_y(x,y,u))$.

Proof. Rewriting Equation 3.5.2 as

$$\frac{\partial \delta_\infty}{\partial y} + \frac{q_x(x,y,u)}{q_y(x,y,u)}\frac{\partial \delta_\infty}{\partial x} + \frac{p(x,y,u)}{q_y(x,y,u)}\delta_\infty + \frac{I_h^\infty R_h}{q_y(x,y,u)} = 0. \tag{3.5.4}$$

The residual vector can be written as a summation of bump functions given by

$$I_h^\infty R_h = \sum R_{h,i} H_{v_i},$$

where $R_{h,i}$ is the component of residual vector in control volume i and H is the Heaviside distribution around V_i. Due to the linearity of the above equation, the solution of Equation 3.5.4 can be written as the superposition of the solutions of subproblems that satisfy

$$\delta_i + \frac{R_{h,i}}{p(x,y,u)} = 0, \quad (x,y) \in V_i, \tag{3.5.5}$$

and

$$\frac{\partial \delta_i}{\partial y} + \frac{q_x(x,y,u)}{q_y(x,y,u)}\frac{\partial \delta_i}{\partial x} + \frac{p(x,y,u)}{q_y(x,y,u)}\delta_i = 0, \quad \text{otherwise}, \tag{3.5.6}$$

where $i = 1, ..., N_{V_i}$, such that N_{V_i} is the number of control volumes. Solution to Equation 3.5.6 can be obtained using the Method of Characteristics. Suppose we can find a

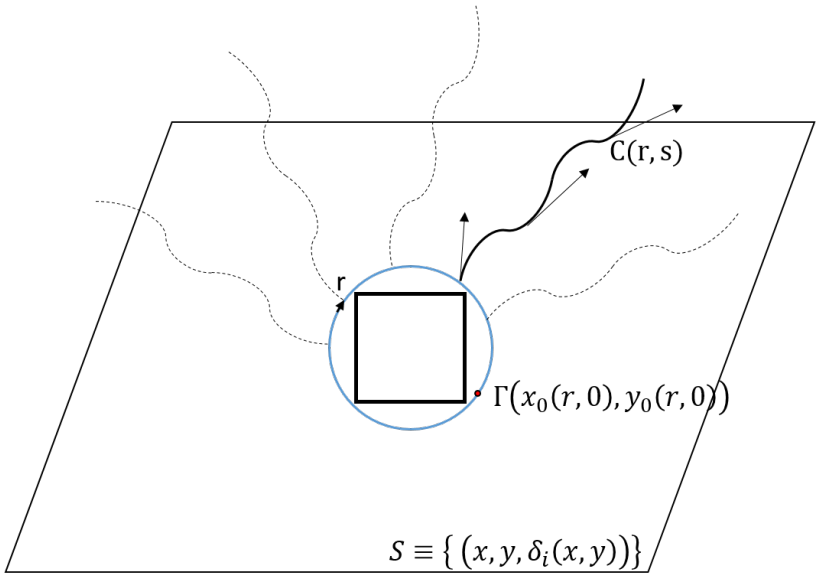


Figure 3.8: Gamma curve, $\Gamma(x_0(r), y_0(r)))$ for a Cartesian two dimensional problem. A circumscribing circle is assumed to be the gamma curve for applications as smoothness of the gamma curve is desirable.

solution $\delta_i(x,y)$, then this function can be represented as

$$S \equiv \{x, y, \delta_i(x,y)\},$$

as shown in Figure 3.8. Hence, solving for δ_i is equivalent to finding the surface S, generated

by δ_i. If δ_i is the solution at (x, y), then

$$\left(\frac{q_x(x,y,u)}{q_y(x,y,u)}, 1, \frac{p(x,y,u)}{q_y(x,y,u)}\right) \cdot \left(\frac{\partial \delta_i}{\partial x}, \frac{\partial \delta_i}{\partial y}, \delta_i\right) = 0.$$

Because $\left(\frac{\partial \delta_i}{\partial x}, \frac{\partial \delta_i}{\partial x}, -\delta_i\right)$ is normal to S at point (x, y), the coefficients in Equation 3.5.6 are perpendicular to the normal and hence these coefficients lie on the tangent plane to S at $(x, y, \delta_i(x, y))$. The integral surface is the union of curves which satisfy this property on S. The initial condition is shown in Figure 3.9. Parameterizing any curve $C \subset S$, by a variable r, we see that we are looking for a curve $C = \{x(r, s), y(r, s), z(r, s)\}$ such that the characteristic equations become

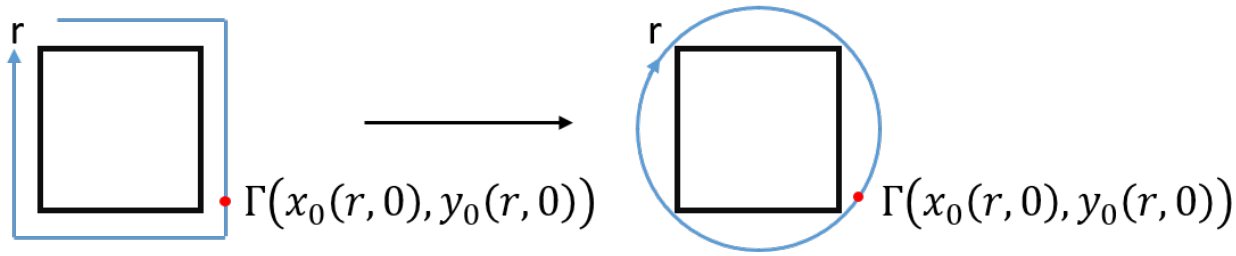


Figure 3.9: Gamma curve, $\Gamma(x_0(r), y_0(r)))$ for a two dimensional Cartesian problem. A circumcircle is assumed to be the gamma curve for applications as smoothness of the gamma curve is desirable.

$$\begin{aligned}
\frac{dy_1}{ds}(r,s) &= 1 \qquad y_1(r,0) = y_0(r) \\
\frac{dx_1}{ds}(r,s) &= \frac{q_x}{q_y}(r,s) \qquad x_1(r,0) = x_0(r) \\
\frac{dz_1}{ds}(r,s) &= -\frac{p}{q_y}(r,s) z_1(r,s) \qquad z_1(r,0) = -\frac{R_{h,i}}{p(x_0,y_0,u)}.
\end{aligned}$$

Solving the above ODEs, we obtain

$$y_1(r,s) = s + y_0(r) \tag{3.5.7}$$

$$x_1(r,s) = x_0(r) + \int_0^s \frac{q_x}{q_y}(r,s)ds \tag{3.5.8}$$

$$z_1(r,s) = -\frac{R_{h,i}}{p(x_0,y_0,u)} e^{-\int_0^s \frac{p}{q_y}(r,s)ds}. \tag{3.5.9}$$

Similarly, writing the characteristic equations for Equation 3.5.3, we get

$$\begin{aligned}
&\frac{dy_2}{ds}(r,s) = 1 \qquad y_2(r,0) = y_0(r) \\
&\frac{dx_2}{ds}(r,s) = q^* \qquad x_2(r,0) = x_0(r) \\
&\frac{dz_2}{ds}(r,s) = -\gamma^* z_2(r,s) \qquad z_2(r,0) = -\frac{R_{h,i}}{p(x_0,y_0,u)}.
\end{aligned}$$

Solving the above ODEs, we obtain

$$y_2(r,s) = s + y_0(r) \tag{3.5.10}$$

$$x_2(r,s) = x_0(r) + q^* s \tag{3.5.11}$$

$$z_2(r,s) = -\frac{R_{h,i}}{p(x_0,y_0,u)} e^{-\gamma^* s}. \tag{3.5.12}$$

Reiterating the aim of this work, conservative estimates are sought to Equation 3.5.9. To ensure conservative estimates in the sense of support, two conditions are necessary. Comparing Equations 3.5.8 and 3.5.11 and Equations 3.5.9 and 3.5.12, mathematically, the conditions can be written as

1. $q^* s \geq \int_0^s \frac{q_x}{q_y}(r,s)ds$, and
2. $\gamma^* s \leq \int_0^s \frac{p}{q_y}(r,s)ds$.

The first condition ensures that the slope of the homogenized characteristics is greater than the slope of the actual characteristic lines, which means that the homogenized lines possess larger wave speeds. The second condition ensures that the decay of the state variable along the characteristic lines is smaller in the homogenized system than in the case of Equation 3.5.2. Noting that the Newton update for an individual residual entry invariably decays monotonically, the above mentioned conditions are satisfied when

1. $q^* = \max_{x,y\in\Omega} \frac{q_x}{q_y}(x,y)$, and
2. $\gamma^* = \min_{x,y\in\Omega} \frac{p}{q_y}(x,y)$.

Hence, Equation 3.5.3 will result in conservative estimates to Equation 3.5.2, given q^* and γ^* satisfy the two conditions listed above. □

Example 5. *Given Equation 3.5.2 with* $q_x(x,y) = rand(50,100)$, $q_y(x,y) = rand(25,60)$ *and* $p(x,y) = rand(0.5,1.0)$, *where rand is a random number generator function, show that Equation 3.5.3 results in a conservative estimate.*

Solution. We solve the given example on a 10×10 grid with random coefficient values as described in the problem. The first step is to compute the characteristic curves for Equation 3.5.2 which is done by replacing the integration operator with a summation operator such that

$$x_1(r, s_i) = x_0(r) + \sum_{i=1}^{N_{V_i}} \left(\frac{q_x}{q_y}\right)_i (s_{i+1} - s_i), \tag{3.5.13}$$

and

$$z_1(r, s_i) = -\frac{R_{h,i}}{p(x_0, y_0, u)} e^{-\sum_{i=1}^{N_{V_i}} \left(\frac{p}{q_y}\right)_i (s_{i+1} - s_i)}, \tag{3.5.14}$$

where $i = 1, ..., N_{V_i}$. Equation 3.5.13 gives the speed of the wave in each control volume as shown by the solid (blue) lines in Figure 3.10 for the first quadrant. Similar results will be obtained for the complete circular Γ-curve as shown in Figure 3.9. The slope of each line is dictated by the underlying coefficient field. Equation 3.5.14 gives the decay of Newton update along each characteristic curve. This is shown in Figure 3.11 by the solid (blue) contour lines with an initial value, $-\frac{R_{h,i}}{p(x_0,y_0,u)} = 100.0$. The irregularity in the contours show the variable nature of the linear differential equation. Next the estimate is derived by computing the coefficients in Equation 3.5.3 such that

$$q^* = \max_i \left(\frac{q_{x_i}}{q_{y_i}}\right)$$
$$\gamma^* = \min_i \left(\frac{p_i}{q_{y_i}}\right).$$

Using q^*, straight characteristic lines are obtained that represent faster wave speeds. Homogenized characteristic curves are shown in Figure 3.10 by dotted (red) lines. Equation 3.5.12 is solved to obtain the decay of Newton updates along the homogenized curves. Figure 3.11

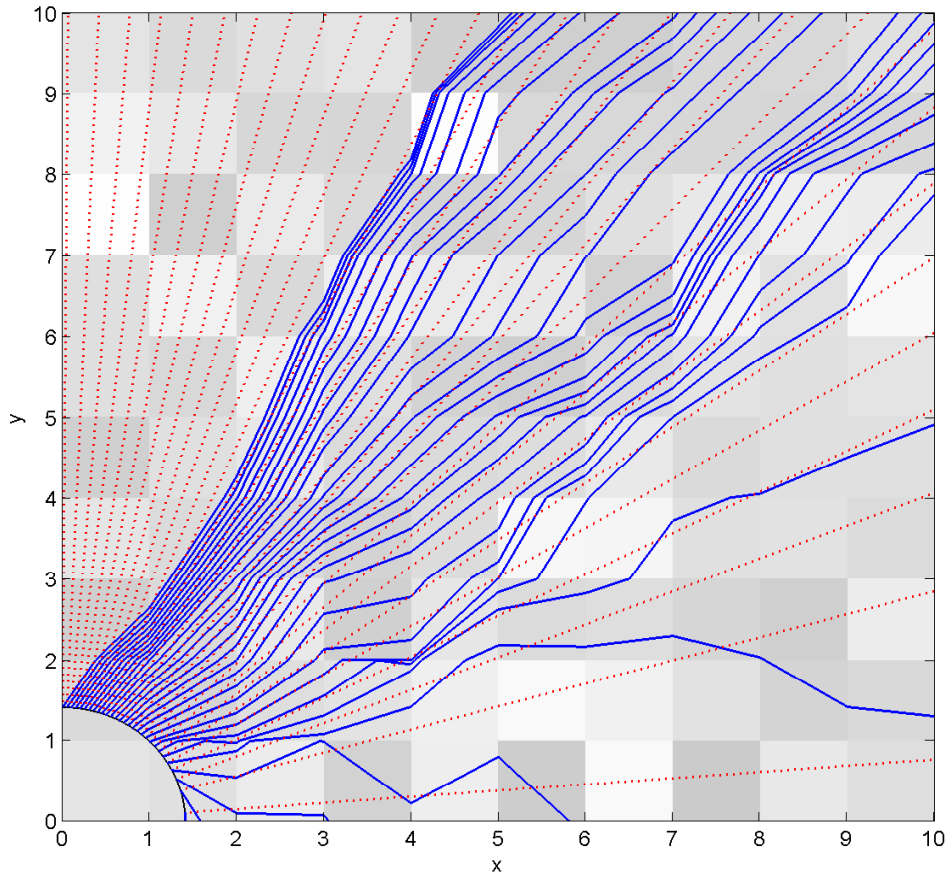


Figure 3.10: Comparison of the homogenized characteristics and the variable coefficient equation solution.

shows the decay contours generated by solving the constant coefficient differential equation represented by dotted (red) lines. It is clearly evident that the homogenized solution decays slower than the solution of Equation 3.5.2 and hence the estimate is conservative.

3.5.2 Approximation 2: spherical symmetry.

Instead of solving a multi-dimensional problem to obtain an estimate, we can recast the problem into a one dimensional equation and compute radial solutions. The following theorem transforms Equation 3.5.1 into a radial ordinary differential equation and provides conditions on obtaining a conservative estimate.

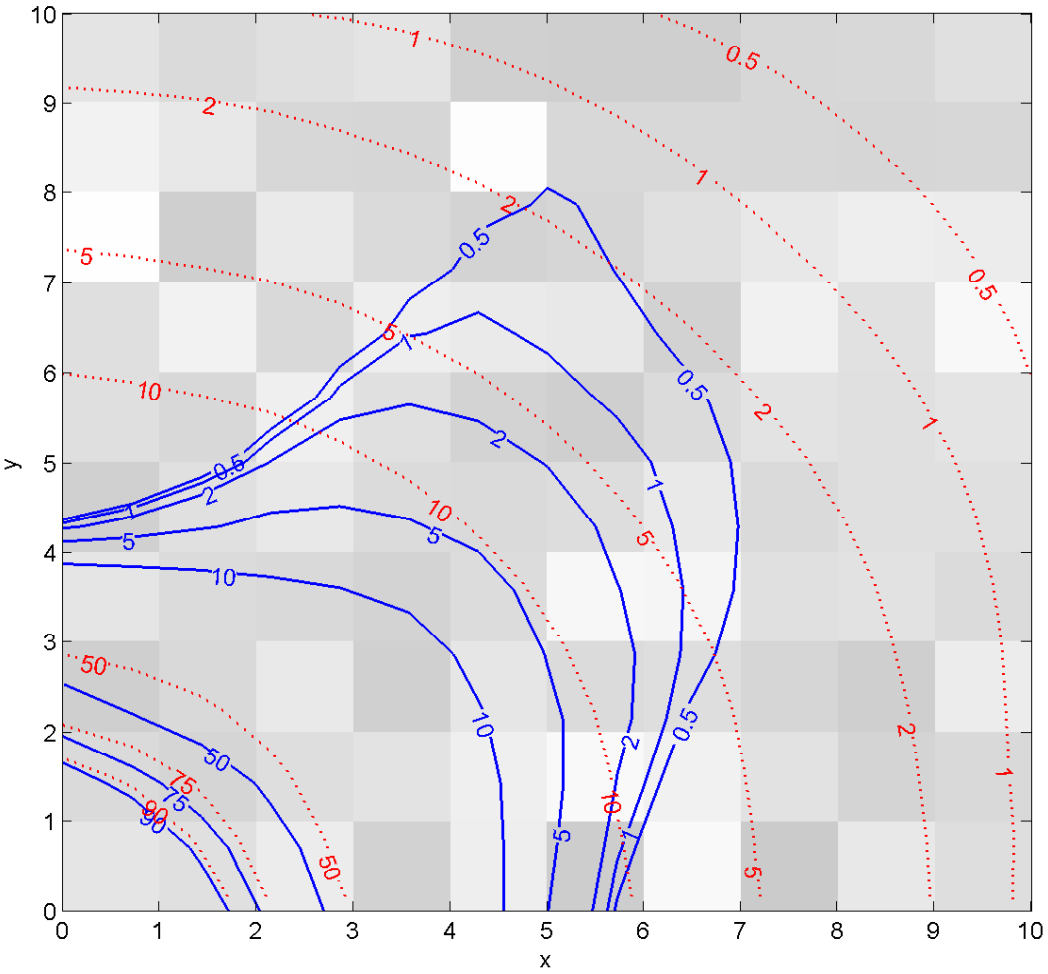


Figure 3.11: Comparison of the homogenized characteristics and the variable coefficient equation solution.

Theorem 6. *For a given first order equation of the form*

$$q_x(x,y,u)\frac{\partial \delta_\infty}{\partial x} + q_y(x,y,u)\frac{\partial \delta_\infty}{\partial y} + p(x,y,u)\delta_\infty + I_h^\infty R_h = 0, \tag{3.5.15}$$

conservative estimates can be obtained by solving

$$\begin{cases} \frac{\partial \delta_i^*}{\partial \xi} + \gamma_\xi^* \delta_i^* & = 0, \quad \xi \in V_i, \\ \delta_i^* + \frac{R_{h_i}}{p(\xi_0,u)} & = 0, \quad \textit{otherwise}, \end{cases} \tag{3.5.16}$$

such that

$$\xi := \begin{cases} x, & \textit{if } \max(q_x) \geq \max(q_y) \\ y, & \textit{otherwise.} \end{cases} \tag{3.5.17}$$

Proof. Equation 3.5.15 is linear and multidimensional. As discussed in the previous section, superposition may be applied by decomposing the projected residual into a sum of local bump functions that are piecewise nonzero over each control volume in the mesh. That is, let V_i denote one control volume in Ω_h with $i \leq \|\Omega_h\|_0$. For each i, we may shift the origin to the centroid of V_i. A conservative estimate is derived in Theorem 5 given by

$$z_2(r,s) = -\frac{R_{h,i}}{p(x_0,y_0,u)} e^{-\gamma^* s}, \tag{3.5.18}$$

where the parameter s can be written in terms of either of the spatial variables, x or y. The term in the exponent, p^*s can be written as $\gamma^*(y_2(r,s) - y_0(r))$ or $\frac{\gamma^*}{q^*}(x_2(r,s) - x_0(r))$, depending on the choice of the variable. For the definitions of the variables used here, please refer to Theorem 5. Recalling from the previous section, $\gamma^* = \min(p(x,y,u)/q_y(x,y,u))$ and hence $\gamma^*/q^* = \min(p(x,y,u^\nu)/q_x(x,y,u^\nu))$. Therefore Equation 3.5.18 becomes

$$z_2(r,s) = -\frac{R_{h,i}}{p(x_0,y_0,u)} e^{-\min(p(x,y,u)/q_\xi(x,y,u))(\xi_2(r,s)-\xi_0(r))}, \tag{3.5.19}$$

where $\xi = x, y$. Subsequently, the solution of Equation 3.5.16 is given by

$$\delta_i^* = -\frac{R_{h,i}}{p(\xi_0,u)} e^{-\gamma_\xi^*(\xi_2-\xi_0)}. \tag{3.5.20}$$

Comparing Equations 3.5.19 and 3.5.20 we can conclude that

$$\gamma^* = \min(p(x,y,u)/q_\xi(x,y,u))$$

where if $\max(q_x) \geq \max(q_y)$, $\xi := x$ will produce conservative estimates and on the other hand, if $\max(q_y) \geq \max(q_x)$, $\xi := y$ will result in a larger support. □

Figure 3.12 shows the results of solving the radial equation as described above for two different cases. Figure 3.12a is obtained by solving Equation 3.5.20 where the difference in the value of the coefficients is small. Blue lines are obtained by solving the variable coefficient two dimensional problem. On the other hand, red (dotted) circles are a result of the solution of Equation 3.5.20. It can be concluded from the contour values that the decay on the radial contours is smaller than the decay on the variable contours. Thus, resulting in a conservative estimate. The case where the preferential direction of flow is in one particular direction is shown in Figure 3.12b. The estimates obtained for this extreme case are conservative.

Expanding this derivation to three dimensions, we will assume that V_i can be approximated by a sphere with a radius $r_{h,i}$. Then Equation 3.5.1 may be approximated by the superposition of the solutions to the sequence of problems,

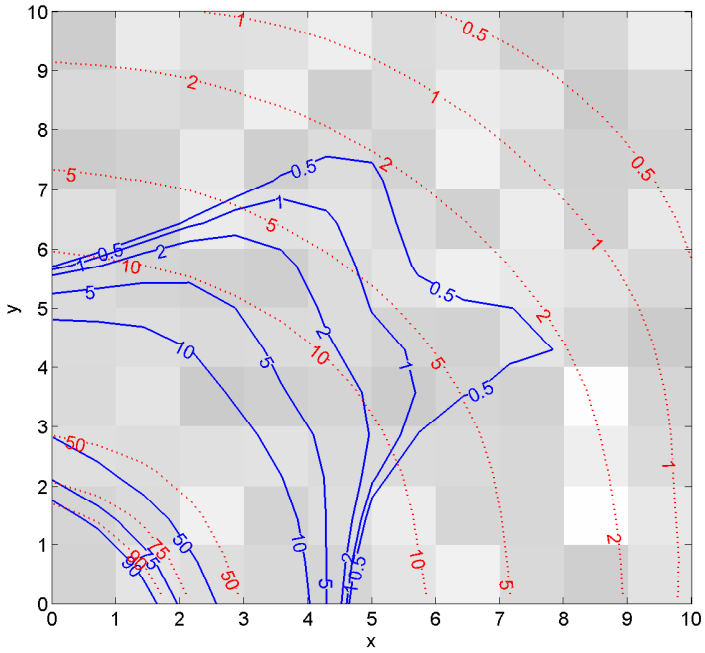
$$\begin{cases} \frac{\partial \delta_i^*}{\partial r} + \gamma^* \delta_i^* & = 0, \quad r \geq r_{h,i}, \\ \delta_i^* + \frac{R_{h,i}}{p\left(r, U_{h,i}\right)} & = 0, \quad r = r_{h,i}, \end{cases} \tag{3.5.21}$$

where now each subproblem involves an internal boundary condition. That is, we have transformed the subproblems over a sequence of associated domains, $\Omega_h - V_i$. For each subproblem, an internal boundary condition on ∂V_i is imposed by applying the solution derived in [65].

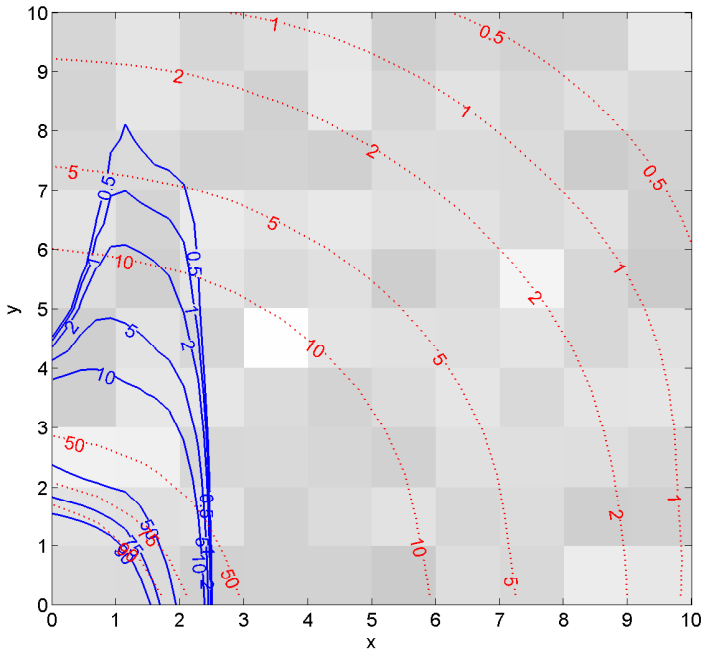
The solutions to subproblems in Equation 3.5.21 are obtained by direct integration,

$$\delta_i^* = -\frac{R_{h,i}}{p\left(r_{h,i}, U_{h,i}\right)} \exp\left(-\gamma^*(r - r_{h,i})\right). \tag{3.5.22}$$

Note that the estimate is monotonically decaying with r given that γ^* is non-negative. The complexity of computation to resolve Equation 3.5.22 over a set of N control volumes given NNZ nonzero residual terms is $O(N * NNZ)$. By exploiting the monotonicity of the above equation, the next section introduces an alternate method that reduces the complexity of support computation to a constant.



(a) $q_x(x, y, u) = rand(50, 100)$ and $q_y(x, y, u) = rand(25, 60)$.



(b) $q_x(x, y, u) = rand(200, 500)$ and $q_y(x, y, u) = rand(25, 60)$.

Figure 3.12: Figures (a) and (b) show the comparison of the radial solution for two cases to the solution of the variable coefficient ODE. Case (a) has comparable q_x and q_y values while in Case (b) there is preferential direction of flow in the y-direction.

3.5.3 The radius of the nonzero support set.

Similar to the pressure derivation, an alternate algorithm is introduced here which reduces the complexity of the algorithm. Because of the decaying character of Equation 3.5.22, we can compute the radius centered at the nonzero residual, $R_{h,i}$, such that the Newton update, δ^*, has sufficiently decayed in magnitude. A trivial sufficient decay condition is the machine precision number. Equating the Equation 3.5.22 to the machine precision, the radius can be obtained for each non-zero residual entry till which it produces a Newton update greater than the machine precision in the domain. The resulting equation can be written as,

$$-\frac{R_{h,i}}{p\left(r_{h,i}, U_{h,i}\right)} \exp\left(-\gamma^*(r - r_{h,i})\right) = \epsilon_s. \tag{3.5.23}$$

Subsequently, solution for the radius of the non-zero support can be obtained by taking the logarithm on both sides of the above equation. Logarithmic transformation is not defined for negative numbers and for negative Newton updates, ϵ_s should have a negative value. To make the derivation consistent and to allow for the logarithmic transformation, we take the absolute value of the term outside the exponent in Equation 3.5.23 and consider $\epsilon_s > 0$. This results in

$$r = r_{h,i} - \frac{1}{\gamma^*} ln\left(abs\left(\frac{p\left(r_{h,i}, U_{h,i}\right)}{R_{h,i}}\right)\epsilon_s\right) \tag{3.5.24}$$

3.6 Algorithm

This section presents the algorithms for the previously developed mathematical framework. Schematically this process is illustrated in Figure 3.13.

First the residual vector is inspected for values that will result in nonzero Newton updates using the diagonal entries in the Jacobian. Due to the linearity of the differential equation, the problem is decomposed into several isolated residual bump functions as stated in Equation 3.3.1. For each isolated residual value, Figure 3.13a, the problem is solved in the continuous space and the radius is calculated as shown in Figure 3.13b. The flagged region is given by the circle centered at the isolated residual location with a radius given by the

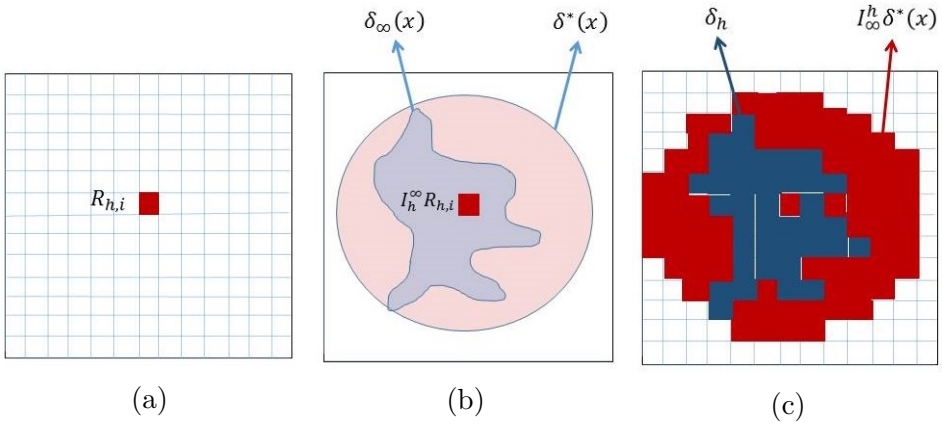


Figure 3.13: A schematic of the conservative estimation process for the Newton update for one nonzero residual entry: (a) An isolated nonzero entry in the residual; (b) The nonzero residual is taken as a piecewise constant bump function. The exact infinite-dimensional Newton update (blue) is contained within the radial conservative estimate (red); (c) The radial infinite-dimensional estimate is restricted onto the mesh (dark red) and encloses the numerical Newton update (dark blue).

formulas developed in the previous sections. This flagged domain is then projected onto the discrete space as shown in Figure 3.13c and a reduced linear system is solved.

Next, we present the pseudo codes for the localized linear solver developed in this work. Algorithm 1 gives details of the localized Newton solver which takes as input a state at time level n and outputs the updated state values. The global residual is calculated only at the beginning of the simulation and thereafter local residuals are calculated accompanied by a local update in the global residual. Given a residual vector, Line 3 in Algorithm 1 determines the residual active set, details of which are given in Algorithm 2. $\delta_{h,i}^{0,i}$ refers to the Newton update caused by an isolated residual entry, i, in the ith gridcell. This results in a better criteria to determine the active set. If the absolute value of this update is greater than ϵ, the residual entry is considered active. The value of ϵ differs for the flow and transport solve. For flow, $\epsilon_p = 0.00001$ and for transport, $\epsilon_s = 0.000001$. The next step is to flag the area contained within the circle of radius calculated for each nonzero residual entry. The DO_FLAGGING routine is outlined in Algorithm 3. $\mathbf{M}$ is the maximum size (in percent of the entire domain) of the active set for which we perform localization. If the percent nonzero entries in the active set is more than $\mathbf{M}$, typically 60%, we solve the full Newton

system. If the support of the active set is less than **M**, then for each element in I_{active}, the radius of effect is calculated using equation 3.4.27 for flow and Equation 3.5.24 for transport. The total area that needs to be solved is given by the union of the circles with radius r, calculated in the previous step. Localization routine provides the subset of the area of the domain that needs solution for the subsequent Newton iteration. The localized linear system is then solved in Line 6. The local residuals are reevaluated on this subset followed by a local update in the global residual. This process is repeated until convergence of flow and transport.

Algorithm 1: LOCAL_NEWTON

input : U^n
output: U^{n+1} such that $R_h(U^{n+1}; U^n, \Delta t) \approx 0$

1 $I_{active} \leftarrow \{\}$;
2 Compute R_h on the mesh;
3 $I_{active} \leftarrow$ DETERMINE_ACTIVE$(R_h, 1, 2, ...N)$;
4 $I_{active} \leftarrow$ DO_FLAGGING(I_{active});
5 **while** *not converged** **do**
6 Solve $J(I_{active})\delta_{h,active} = -R(I_{active})$;
7 Update $[U]^{\nu+1}_{active} = [U]^{\nu}_{active} + \delta_{h,active}$;
8 **if** $\|I_{active}\|_0 < M$ **then** $I_{active} = I_{active} \cup I_{neighbor} * *$;
9 Compute $R_h(I_{active})$;
10 $I_{active} \leftarrow$ DETERMINE_ACTIVE(R_h, I_{active});
11 $I_{active} \leftarrow$ DO_FLAGGING(I_{active});
12 Update global R_h;

Algorithm 2: DETERMINE_ACTIVE

input : R_h, I^{IN}_{active}
output: I^{OUT}_{active}

1 $I^{OUT}_{active} \leftarrow \{\}$;
2 **for** $i \in I^{IN}_{active}$ **do**
3 Compute $\delta^{0,i}_{h,i}$;
4 **if** $|\delta^{0,i}_{h,i}| > \epsilon_m$ **then**
5 $I^{OUT}_{active} = I^{IN}_{active} \bigcup i$;

Algorithm 3: DO_FLAGGING

input : I^{IN}_{active}
output: I^{OUT}_{active}

$I^{OUT}_{active} \leftarrow \{\}$;
if $\|I^{IN}_{active}\|_0 < M$ **then**
  for *each* $k \in I_{active}$ **do**
    Calculate r_k;
    for $j \in B(x_k, r_k)$ **do**
      $I^{out}_{active} = I^{out}_{active} \bigcup j$
else $I^{OUT}_{active} \leftarrow \{1, 2, ...N\}$;

Subroutine	LOCAL_NEWTON	FULL NEWTON
Solve	M_1^β	N^β
Update δ	M_1	N
Compute R_h	M_2	N
DETERMINE_ACTIVE	M_2	-
DO_FLAGGING	$\theta * M_2 * \Lambda$	-
Update R_h	M_3	-

Table 3.1: Complexity analysis for LOCAL_NEWTON

3.6.1 Complexity analysis

In this section we study the complexity of each of the sub-algorithms presented above. Table 3.1 summarizes the complexity comparison between the local Newton algorithm and the full Newton's method. Here, M_1 is the initial support that needs to be solved and M_2 is the updated set of residuals. We can readily conclude that the $M_1 \geq M_2$ and hence the worst case scenario would be $M_1 = M_2$. M_3 is the support set over which the residual must be updated which is bounded above by M_1. For Newton's method to converge, the support of the flagged domain over several iterations must shrink and hence $\theta < 1$. Λ is the average area of the domain flagged by each residual entry in the active set. From the table, the complexity of the full Newton's method can be derived as

$$2 * N + N^\beta.$$

On the other hand, considering the worst case scenario for the local Newton's method, we

obtain the complexity to be

$$4 * M_1 + M_1^{\beta} + \theta * M_1 * \Lambda.$$

In the above equations, β is the exponent of complexity of the linear solver. The value of β varies with different linear solvers and can be calculated by recording the time of each linear solve for various model sizes. Here, we consider three linear solvers, one direct (PARDISO) and two iterative (ILU0-GMRES and AMG-GMRES). Figure 3.14 results in the

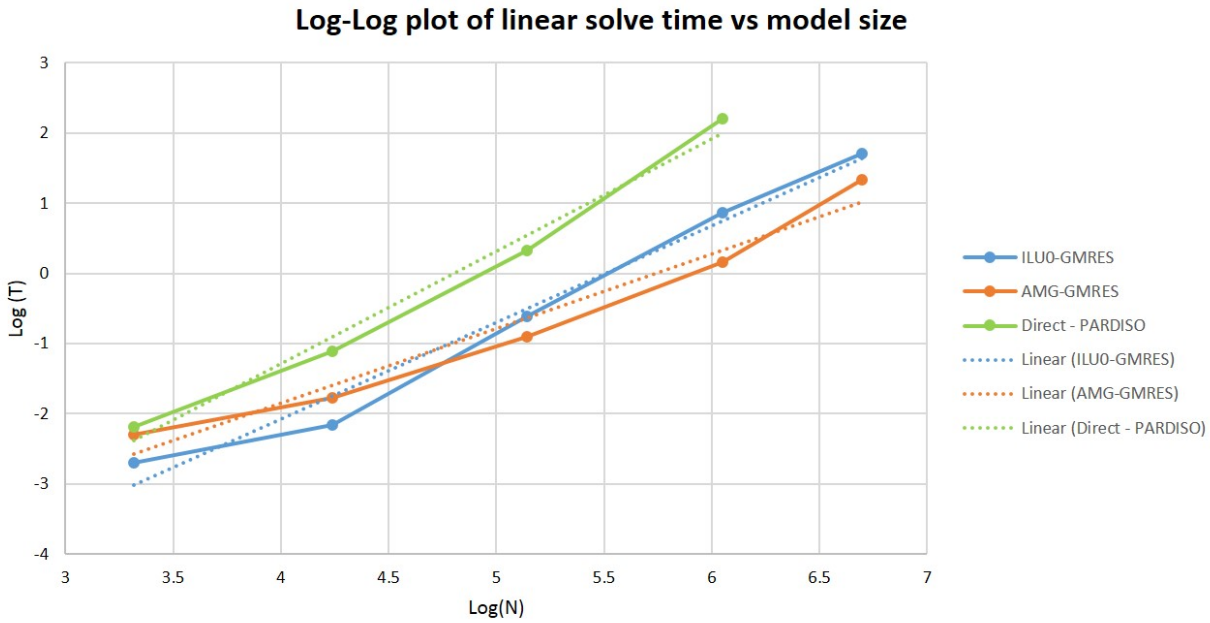


Figure 3.14: Determination of β for PARDISO, ILU0-GMRES and AMG-GMRES.

worst case β value of 2.0 for PARDISO, 1.6 for ILU0-GMRES and 1.3 for AMG-GMRES. Depending on the hardware configuration and the model sizes, these values might change slightly. The dominating term in the complexity of the full Newton's method is N^{β}. In the case of localized Newton's method M_1^{β} or $\theta * M_1 * \Lambda$ can be the dominating term depending on the value of β, θ and Λ. Numerical results are provided in the next section.

CHAPTER 4

RESULTS

4.1 Locality within Newton processes.

The locality within a Newton process for the solution of a time-step is equivalent to the sparsity of the Newton updates over the corresponding number of iterations. The level of locality that is present is related to, but quite distinct from the locality that will be present over the entire time-step. The time-step update is the sum of all Newton updates in the iteration. While the time-step update may be dense, the individual Newton updates may show a relatively large degree of sparsity. To better illustrate this, we consider an example. Figure 4.1 shows a heterogeneous permeability field in which two water injection

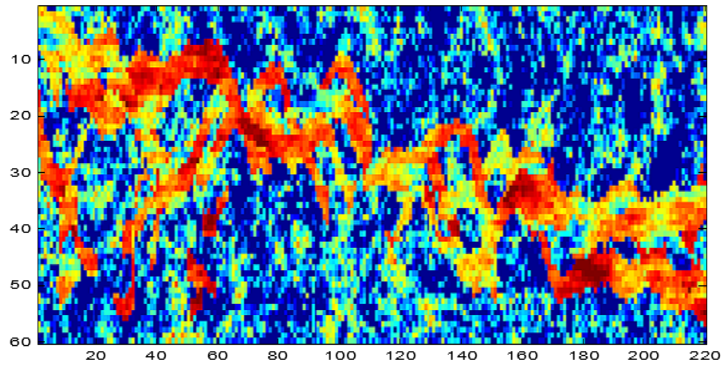
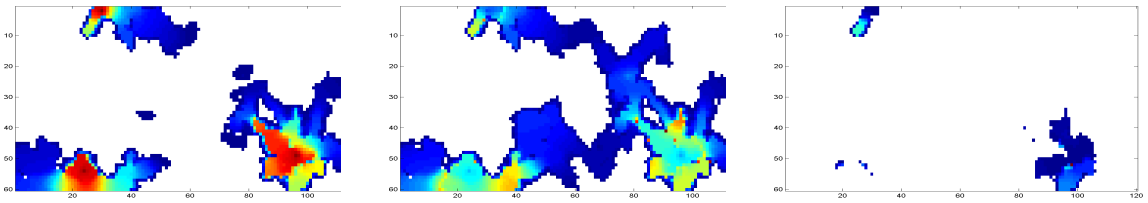


Figure 4.1: Permeability field (log-scale) of a slice of the SPE10 comparative study model ([14]) with 60×220 gridblocks.

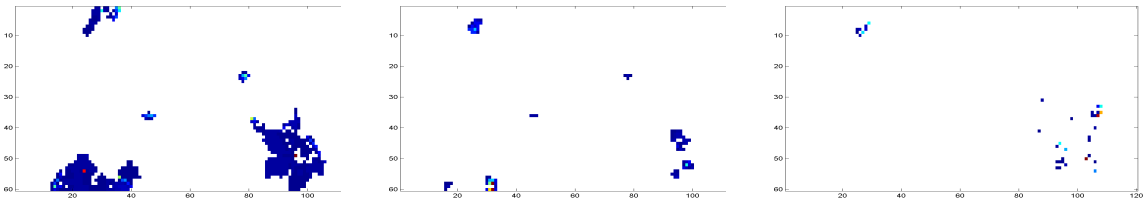
wells and three oil production wells are operated. Small rock compressibility is used and the oil and water densities are functions of pressure. Initially, pressure and saturation fields are assumed to be uniformly distributed. The sequential implicit solution for one time-step is obtained. Figures 4.2 and 4.3 show plots of the absolute values of the Newton

updates for pressure and saturation, respectively. The warm colors show a larger update while the white spaces indicate insignificant change. For some cases, different number of Newton steps might be required for the two systems to converge to their respective solutions as the convergence rate depends on the specific problem being solved. In this case, more iterations were required to converge the saturation for the time-step, whereas the pressure field converged after three iterations. Clearly, over this time-step, the pressure evolution



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Figure 4.2: Pressure updates for two Newton iterations. The old pressure state is uniformly distributed



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3.

Figure 4.3: Saturation updates over three consecutive Newton iterations.

(Figure 4.2) is less local. Often, the sum of the updates span the entire mesh, resulting in a global change for the time-step. Notice however, that there is substantial locality within individual iteration updates. Moreover, the sparsity of the updates is rather complex, and it is generally difficult to predict solely based on well controls and their locations. The saturation updates (Figure 4.3) on the other hand exhibit a great degree of locality, and they appear to progress with a traveling wave character that is shaped by the underlying total velocity field. The precise local velocities of the Newton updates are difficult to predict.

The simulation time-step was repeated except that at the first Newton iteration, a few nonzero updates were neglected. The convergence rate for both pressure and saturation

degraded considerably. By truncating 2% of the pressure updates in the first iteration, the number of iterations required for convergence doubled. By omitting 2% of the nonzero updates for saturation in the first three iterations, the number of iterations required for convergence tripled.

This example empirically highlights three issues:

1. There is considerable locality (sparsity) within Newton updates for both flow, and transport.

2. The precise location of the updates within a given iteration is difficult to predict.

3. Indiscriminately missing a few nonzero update components can deteriorate the overall nonlinear convergence rate considerably.

We present the computational results for flow and transport in multiple problem settings. The results presented in this section require us to compute the numerical Newton update for each iteration and comparing the results with the solution obtained from the analytical estimates developed in the previous sections. First, computational examples are described followed by some flagging results. Comparison between different homogenization strategies is shown next, which provides justifications for the current choice of averaging strategy. Two important factors that affect the degree of locality are the compressibility and the time step size. In each of the cases presented in the next section, the effect of time step size and compressibility is shown. Computational results are presented next, which comprise of three two-dimensional cases in order of increasing complexity and heterogeneity followed by three-dimensional results for the SPE10 comparative study case. Thereafter, the analytical method presented in this work is compared with other heuristic methods studied in the past. Finally, the algorithm is extended to three-phase flow problems where certain assumptions on the governing equations result in simple decoupling. Once the equations are decoupled, the same formulas that are derived in the previous chapters are applicable for flow and decoupled transport. We then present various two and three dimensional cases for three-phase problems that show promising results.

4.2 Computational examples setup

4.2.1 Case 1: Homogeneous domain

Figure 4.4 describes the homogeneous permeability field along with the sources and sinks. The porosity field is considered constant throughout the domain while the other case specific properties are enlisted in Table 4.1. Two producers and one injector operate at constant bottom hole pressure and constant injection rate, respectively. For further details on the fluid properties and model description, please refer to Appendix A and B.

Phase	Oil-water
Permeability range (mD)	5.23 - 20.52
Porosity	0.3
Initial water saturation	0.05
Initial reservoir pressure (psi)	3000.0
Injection rate (stb/day)	1000.0
Well bottom hole pressure (psi)	2500.0
Dimensions ($ft\ x\ ft\ x\ ft$)	1200.0 x 2200.0 x 15.0

Table 4.1: Problem description: homogeneous case

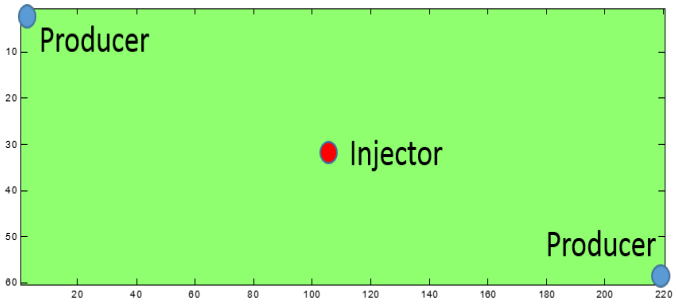


Figure 4.4: Homogeneous permeability field.

4.2.2 Case 2: Gaussian permeability field

Permeability of the first layer of SPE10 model follows a Gaussian distribution. The warm colors represent high permeability regions while the cooler colors show regions of low permeability values. The contrast of permeability in this case is given in Table 4.2 while the well pattern is similar to Case 1. The porosity field is correlated to the permeability field and has a similar structure.

Phase	Oil-water
Permeability range (mD)	0.003 - 4647.5
Porosity	0.05 - 0.4468
Initial water saturation	0.05
Initial reservoir pressure (psi)	3000.0
Injection rate (stb/day)	1000.0
Well bottom hole pressure (psi)	2400.0
Dimensions ($ft\ x\ ft\ x\ ft$)	1200.0 x 2200.0 x 17.0

Table 4.2: Problem description: SPE10 1^{st} layer

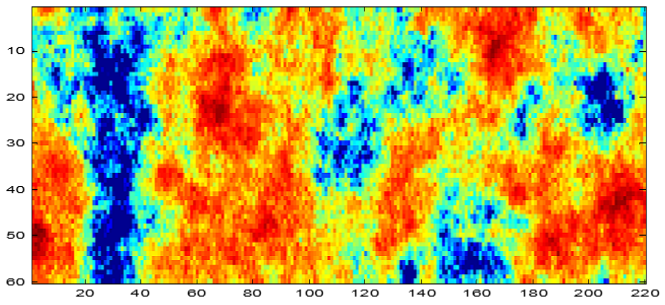


Figure 4.5: Gaussian permeability (Log) field.

4.2.3 Case 3: Channelized depositional environment

Forty-eight layer of the SPE10 geological model simulates a depositional environment which results in a highly channelized geological setting. The permeability contrast is the highest in this case with some regions of higher magnitude throughout the slice.

4.3 Locality

In this section the localization algorithm is implemented in a two-phase immiscible flow simulator for different problem settings. Each case described in the previous section is simulated with different fluid properties and simulation controls. Solutions for the radius of effect given by Equations 3.4.27 and 3.5.24 are used to obtain the flagged domain in the continuous space. Projection of this region onto the discrete space results in the final flagged domain that needs to be solved every nonlinear Newton iteration. Flagging results are shown in Figure 4.7 where the left row represents pressure localization and the right row gives the saturation results. The columns represent three cases described in the previous

Phase	Oil-water
Permeability range (mD)	0.0022 - 20000
Porosity	0.05 - 0.4
Initial water saturation	0.05
Initial reservoir pressure (psi)	3000.0
Injection rate (stb/day)	1000.0
Well bottom hole pressure (psi)	2400.0
Dimensions ($ft\ x\ ft\ x\ ft$)	1200.0 x 2200.0 x 17.0

Table 4.3: Problem description: SPE10 48^{th} layer

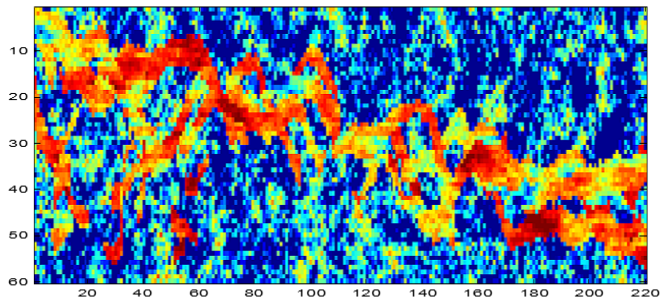


Figure 4.6: Channelized permeability (Log) field.

section. In all the sub-figures shown in Figure 4.7, blue shaded region depicts the actual support of the nonzero (greater than the tolerance) Newton's updates which is obtained by numerically solving a full linear system. The green shaded region represents the area that is flagged by the analytical algorithm developed in this work. The efficiency of the algorithm is dictated by two major factors - the degree of conservativeness and the accuracy of prediction. Any region that is not flagged by the analytical estimate which still produces significant Newton updates will lead to increased number of nonlinear iterations. On the other hand, an excessively conservative flag will reduce the speedup that can be obtained in ideal cases. The controlling parameter behind these factors is the strategy used during the homogenization of the variable coefficient for the second order partial differential equation. The comparison between a few possible strategies is shown in Figure 4.8. It is evident that the infimum of the variable coefficient will always produce conservative solutions and hence has the best accuracy of prediction but in most cases it flags a more global domain. The arithmetic homogenization of the variable coefficient is the least conservative but is unable

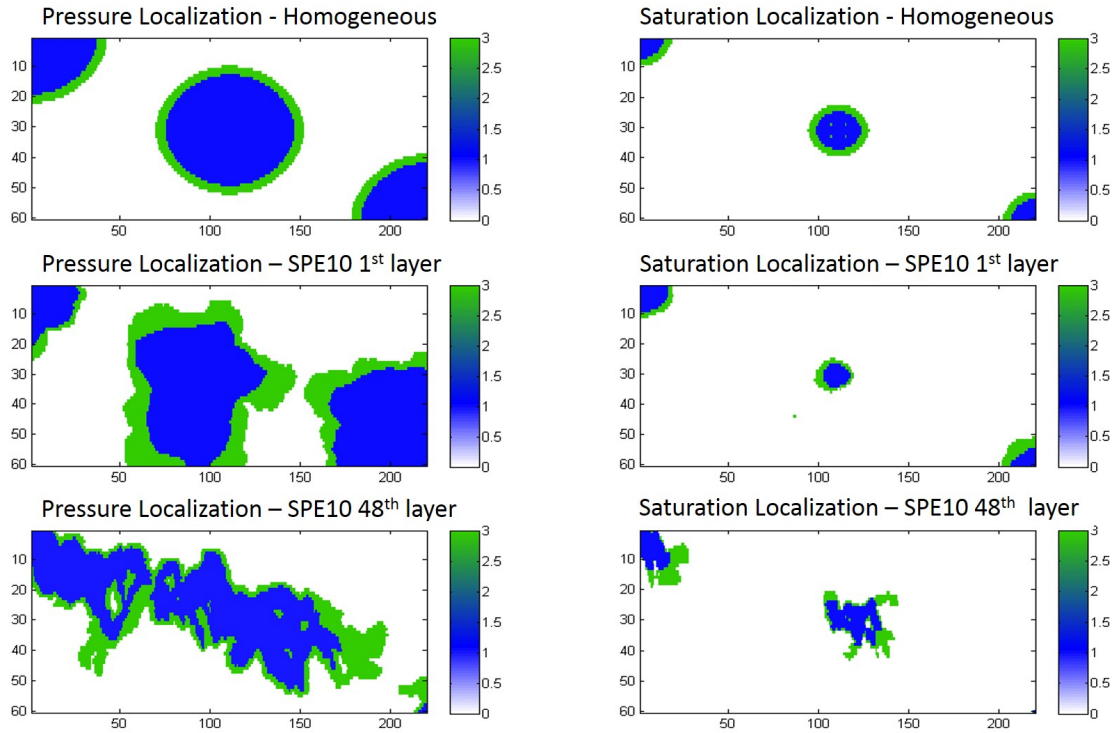


Figure 4.7: Flagging results for pressure (left) and saturation (right) for three cases discussed in the previous section.

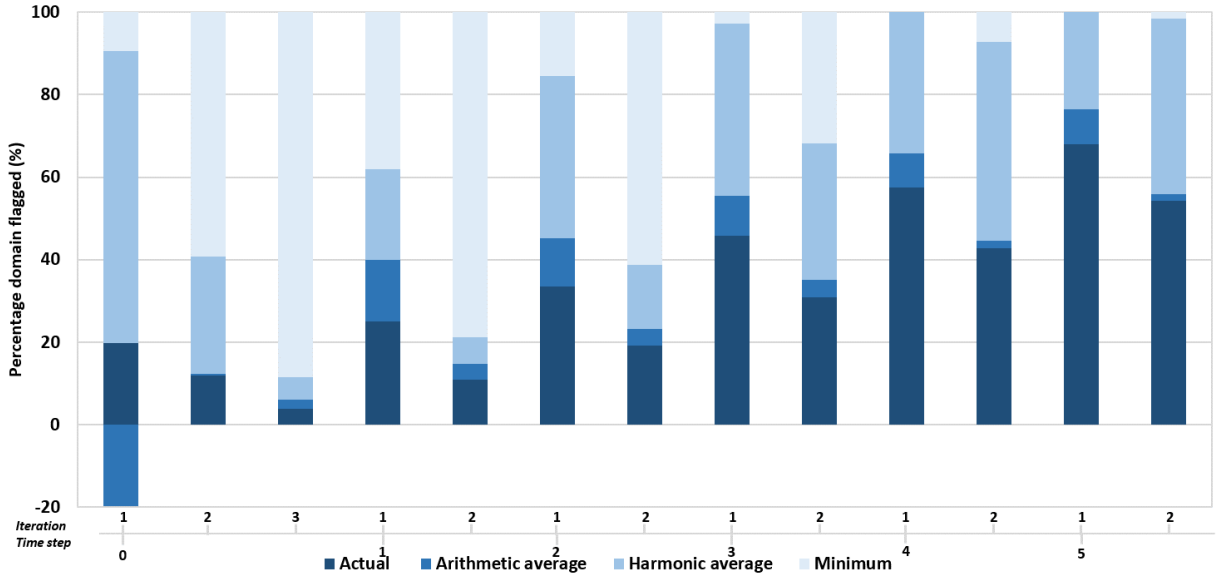
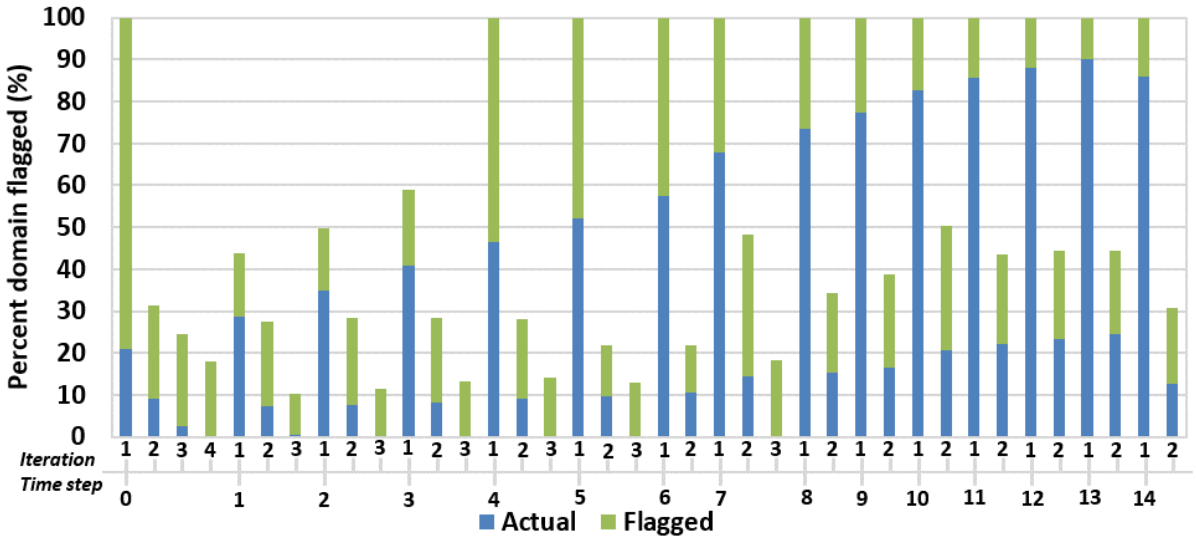
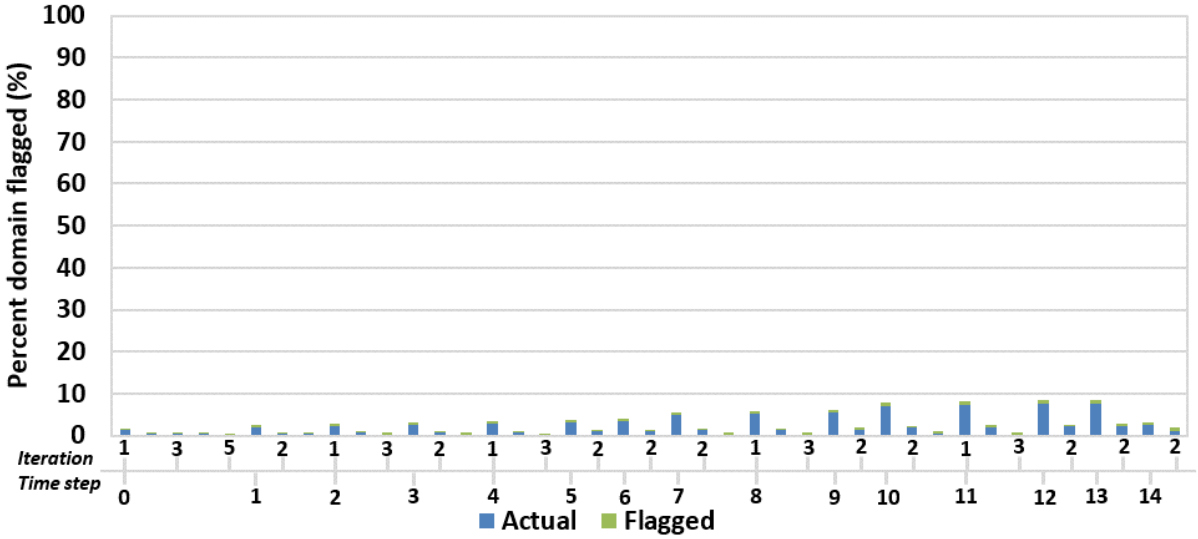


Figure 4.8: Comparison between different homogenization strategies. Infimum produces the most conservative estimates while arithmetic averaging results in the highest speedup.

to flag conservatively for the first iteration of the first time-step. This is due to the steep gradient in the solution of Equation 3.4.27 and very few initial nonzero residual entries.



(a) Localization results for pressure over the course of several time-steps.



(b) Localization results for saturation over the course of several time-steps.

Figure 4.9: Localization results for the case with high oil compressibility over several time-steps. An average domain of 52.58% for pressure (top) and 2.63% for saturation (bottom) is solved over a course of several time-steps.

Comparing the results obtained by using a harmonic averaging strategy, it is conclusive that harmonic averaging is better for the first iteration. It was experimentally observed that using a higher cutoff value for what we consider zero Newton updates for the harmonic strategy, it produces the best results in terms of conservativeness and accuracy. For the results in this chapter a cutoff values of 1.0E-05 for the pressure update and 1.0E-06 for the saturation

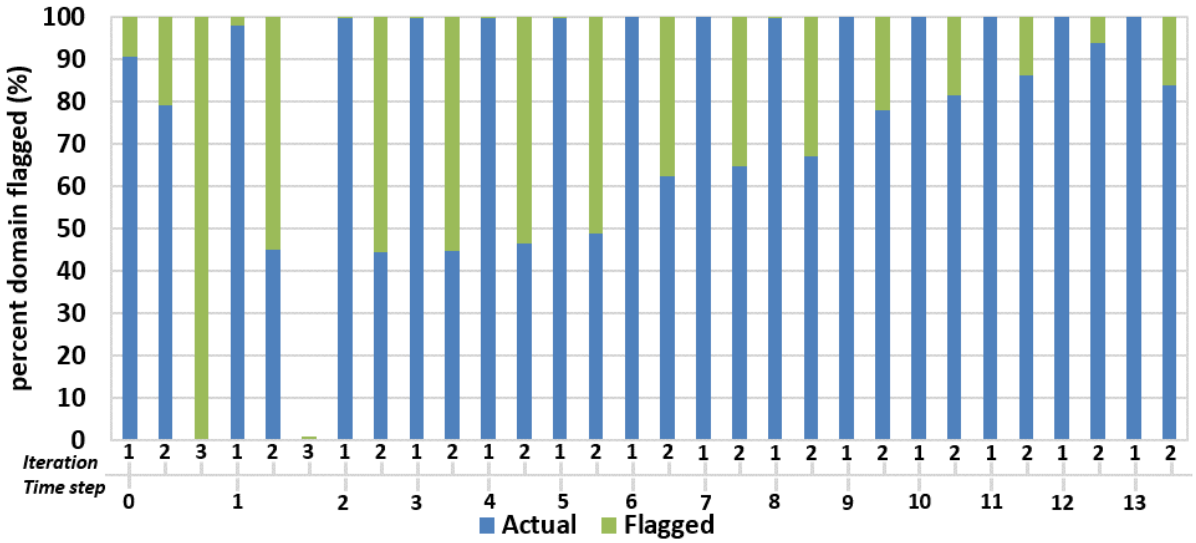
	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.43	0.34	0.31	0.0078	0.0015
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	2.11	2.47	3.09	9.15	181.16

Table 4.4: Run time comparison between the full and localized Newton solvers for the homogeneous permeability case with high oil compressibility.

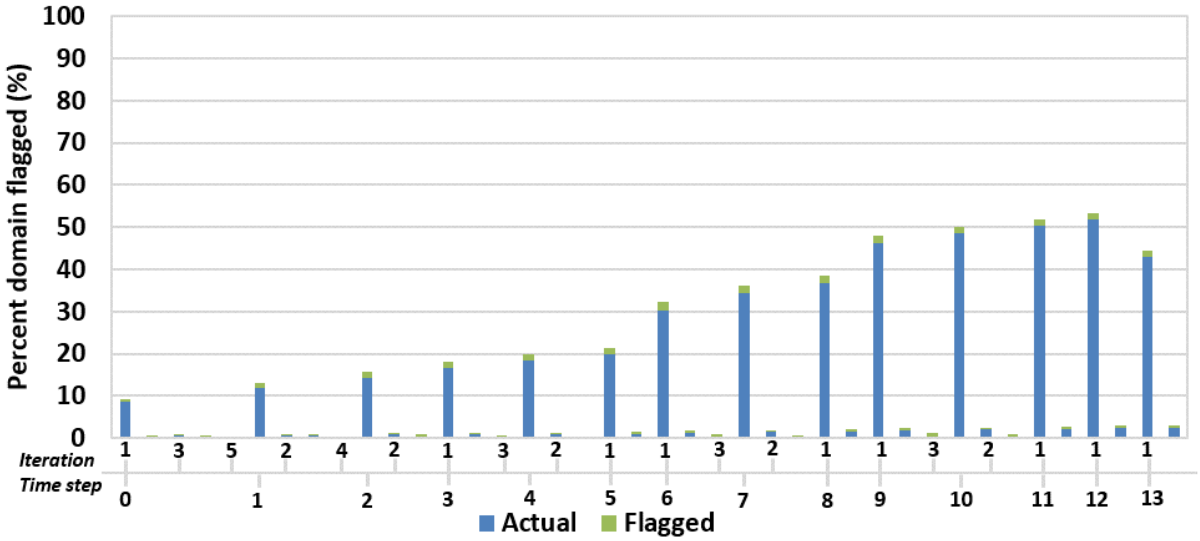
update are used along with harmonic homogenization strategy for the first iteration with a switch to arithmetic homogenization for the rest of the time-steps.

Next, simulation results are provided for each case with varying fluid properties and time step sizes. Figure 4.9 is obtained by the simulation of Case 1 with high oil compressibility and a time-step size of 1 day. Higher fluid compressibility results in smaller pressure diffusion and thus a higher degree of locality in the solution process. In the following figures on locality, the blue bars represent the actual percentage of the domain solved by the simulator while the green bars are the percentage of the domain flagged by the proposed method. The x-axis shows the number of time-steps and the number of iterations in each step. From this figure, it can be inferred that the support of the significant Newton updates is large. Regions with slower physics converge faster and hence as the iterations proceed the locality increases. Regions with fast physics, such as around the well or at the saturation front, require a higher number of Newton iterations to converge to the solution. Due to the high fluid compressibility, the average domain for pressure that needs to be solved every iteration is 52.58%. The saturation propagation is highly localized due to the purely hyperbolic (no capillary pressure) nature of the partial differential equation. In this case, an average domain of only 2.63% needs solution over the course of the simulation.

Table 4.4 shows the comparison between the full Newton and the proposed method for the high compressibility case. Data presented in the table gives the run time for each iteration using full Newton and localized Newton. PARDISO is used to test the direct solver complexity while AMG-GMRES and ILU(0)-GMRES are used to test state of the art iterative solvers. For completeness, along with state of the art iterative solvers, direct solver



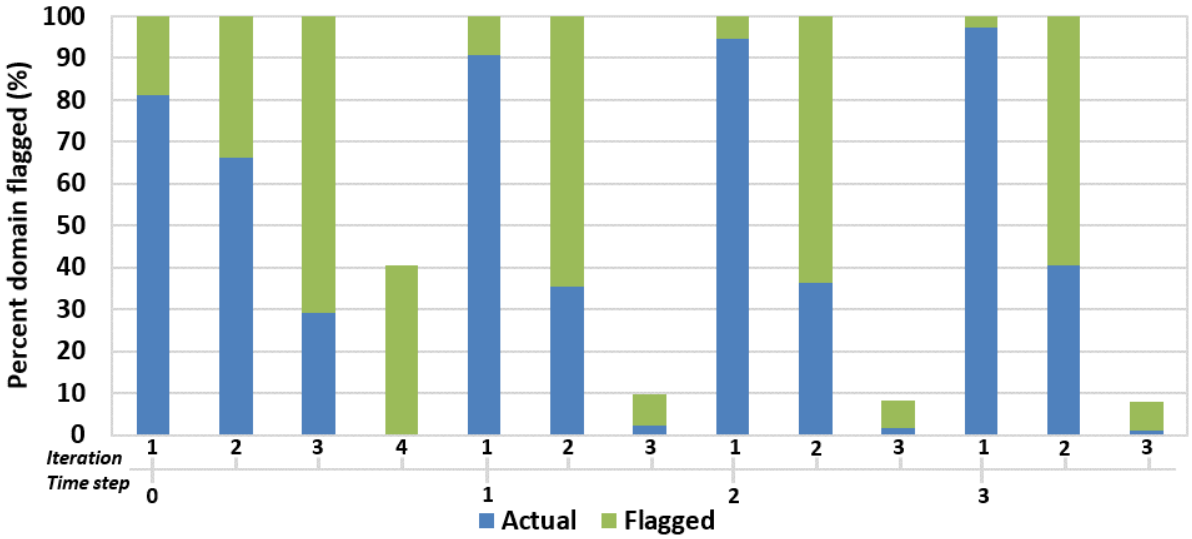
(a) Pressure results for the simulation of Case 1 with low oil compressibility.



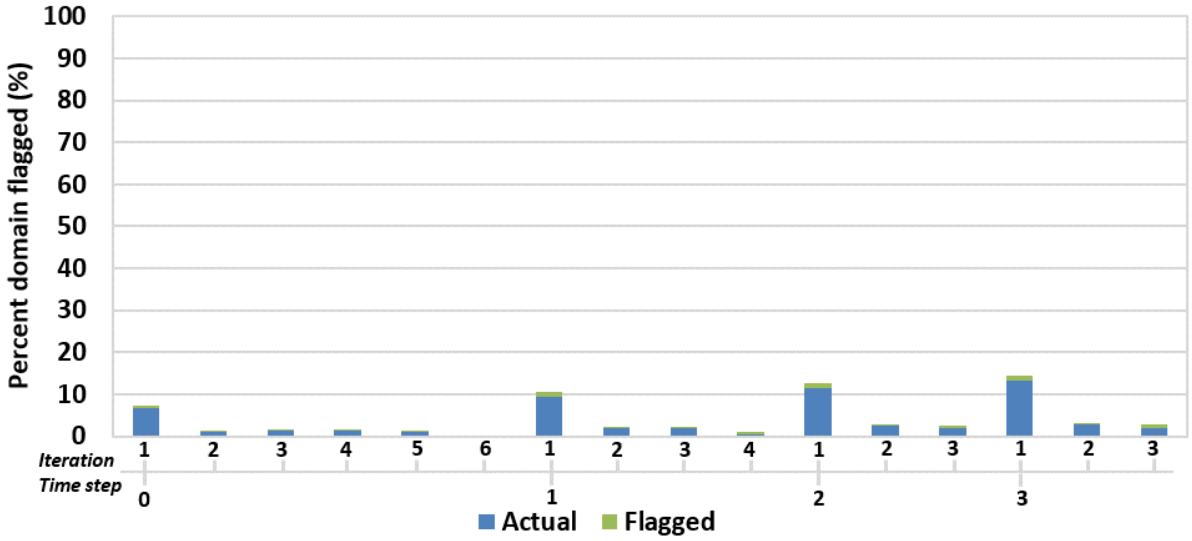
(b) Saturation results for the simulation of Case 1 with low oil compressibility.

Figure 4.10: Localization results for the homogeneous case with different fluid properties and simulation controls.

results are presented in this chapter. In most practical applications, advanced preconditioners with iterative solvers are the main staple. Time taken for the localized solver is the time required to flag the active set and the time required to solve the reduced system. In these results the time saved in the local computation of the residual vector is not taken into consideration for the localized solver. Higher gains can be expected if this is implemented. For the case of high oil compressibility, localized solver is 2.11 times faster than the full



(a) Pressure results for the simulation of Case 1 with big time-steps.



(b) Saturation results for the simulation of Case 1 with big time-steps.

Figure 4.11: Localization results for the homogeneous case with different fluid properties and simulation controls.

Newton solver with AMG-GMRES as the linear solver for the pressure subsystem. The highest speedup obtained was for the direct solver where the speedup is around 3 folds for the pressure system solution and 181 folds for the transport solution. To study the effects of compressibility and time step size on the degree of locality, case 1 is rerun with the same parameters once with a different value of the oil compressibility and once the time step size. Figure 4.10a and 4.10b show the percentage of domain flagged over the course

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.95	0.92	0.92	0.059	0.023
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.02	1.02	1.07	6.11	36.76

Table 4.5: Run time comparison between the full and localized Newton solvers for the homogeneous permeability case with low oil compressibility.

of the simulation for flow and transport, respectively. Due to the low oil compressibility, the pressure diffusion is instantaneous in the bounded domain, resulting in a 96% domain experiencing a nonzero Newton update. Subsequently, due to the global update in the pressure state, underlying velocity field changes and in turn causes a less local saturation update. Although the support of the nonzero Newton update is less local, the magnitude of the saturation update due to the pressure change is small and hence the following iterations are all strictly local. The average domain flagged for the transport part is 12.09%.

Table 4.5 summarizes the run times for each iteration and the speedup obtained by using the localized solver. The speedups obtained are less compared to the high fluid compressibility case due to a bigger support of the Newton update. For problems with bigger initial support than M (60% in this problem), as described in Algorithm 3, the flagging routine will be skipped and the full linear system will be solved which will result in a speedup of 1.0, which effectively refers to the switch to full Newton's method. This switch is an important step as it ensures that in the worst case scenario, the localized Newton method proposed in this work is equivalent to the traditional Newton method. The speedup obtained in this case for the saturation solution for an iterative solver is around 6 times. While in the case of pressure there is hardly any reduction in the linear system size. For the same size of the linear system being solved, a direct solver results in a larger gain due to the high complexity routines.

In the final sub-case, the maximum time-step size is taken to be 32 days. Due to the bigger time-step size, the support of the nonzero Newton update is expect to be larger. It can be observed from Figure 4.11a that the first iteration is always quite global. The

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.67	0.615	0.56	0.01	0.0033
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.41	1.49	1.75	8.49	137.36

Table 4.6: Run time results for the localized linear solver for the two-dimensional homogeneous case with big time-step size

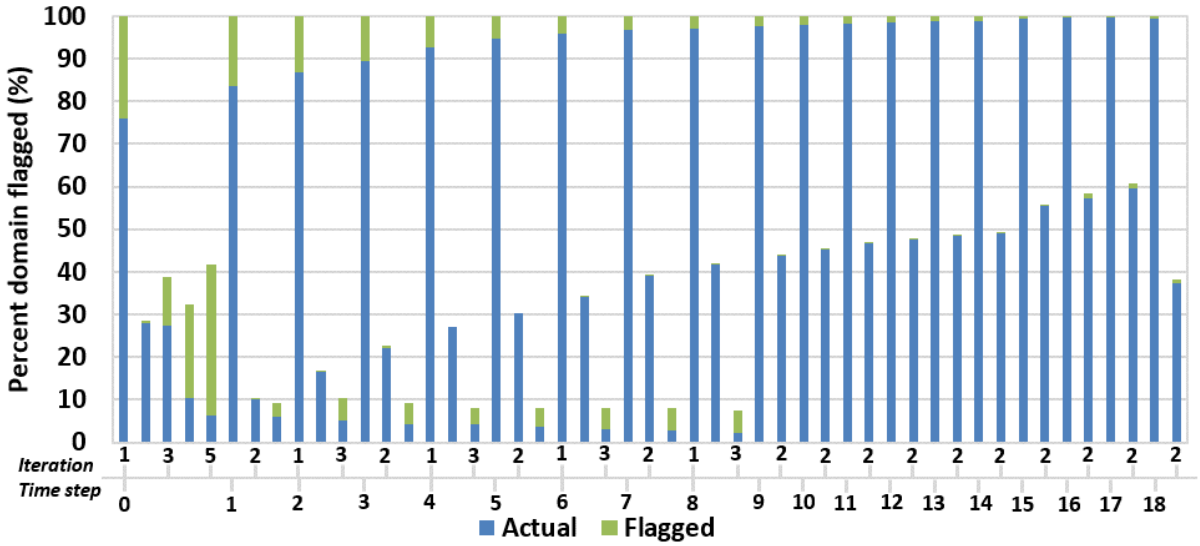
	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.49	0.40	0.36	0.035	0.011
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.89	2.17	2.63	7.16	65.79

Table 4.7: Comparison between localized and full Newton solver in terms of wall clock time for Case 2 (high compressibility).

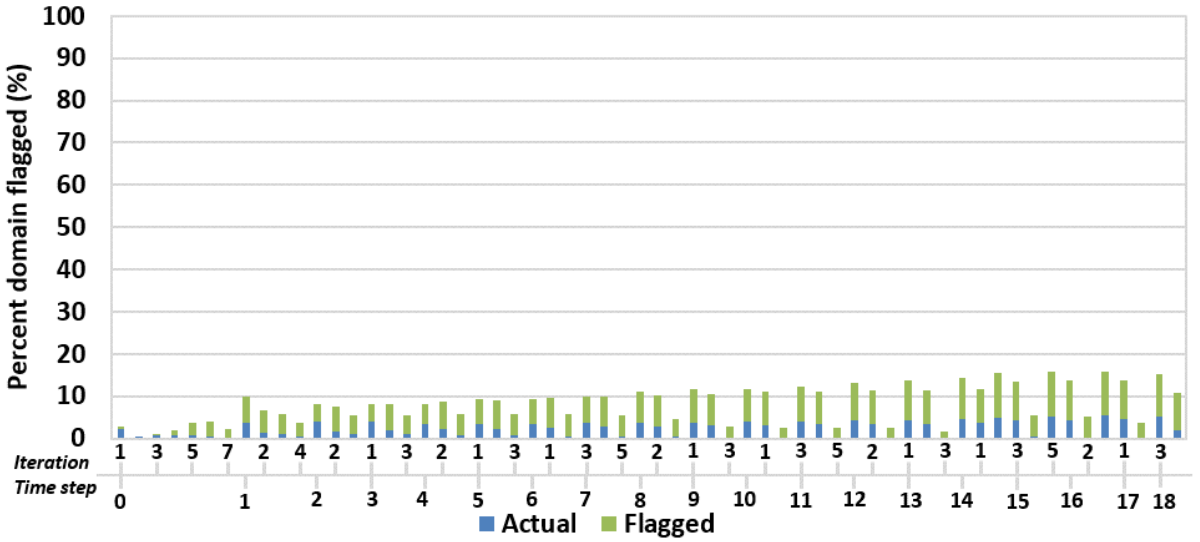
problem is slightly more difficult to solve with big time-steps and hence takes a few more iterations to converge which are all mostly local. On an average around 74% of the domain is flagged over the course of the simulation. For transport, only 4.15% domain needs to be solved to get the same solution as the full Newton's method.

Next, the same problem is tested on a Gaussian permeability and a heterogeneous porosity field. The problem and the well controls are described in Table 4.2. Figure 4.12 shows the localization results for the simulation of two phase flow in the 1st layer of SPE10 comparative study case. On an average, 57% and 8.2% of the domain is flagged for the pressure and saturation equations, respectively. Table 4.7 show the speedups obtained using the proposed method for a high oil compressibility case. Speedup obtained for the case where AMG is used as a preconditioner and GMRES as the linear solver, is 1.89 folds. Similarly, for the direct solver, we obtain 2.63 times faster simulation compared to the solution of the full linear system. In the case of transport, the speedup ranges between 7.16 and 65.79 folds depending upon the type of preconditioner and linear solver used.

The effects of compressibility and time-step size on the performance of the proposed



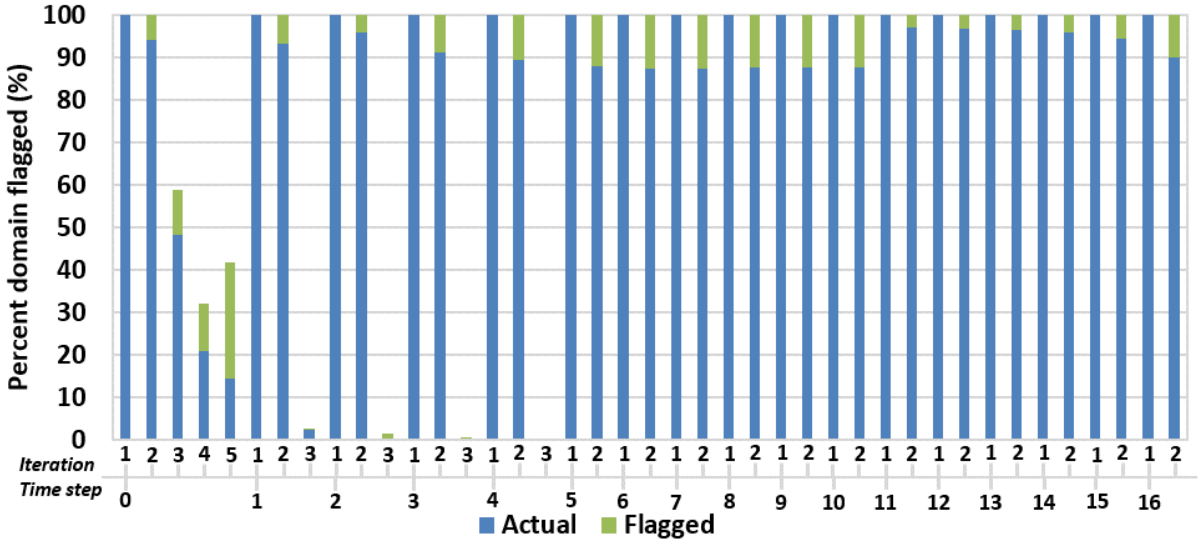
(a) Localization results for pressure over the course of several time-steps.



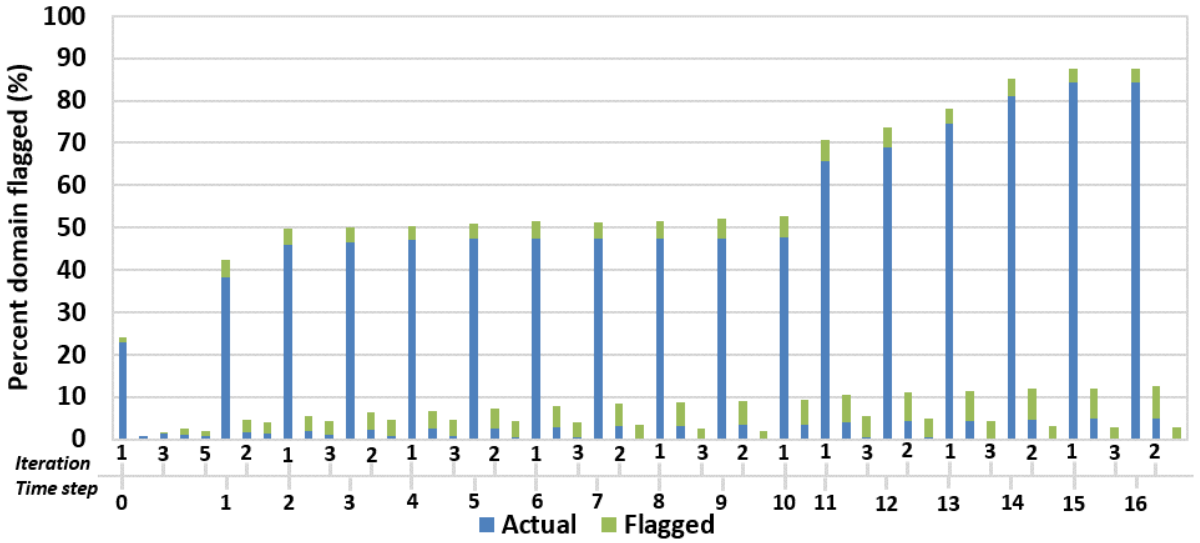
(b) Localization results for saturation over the course of several time-steps.

Figure 4.12: Localization results for the case with high oil compressibility on the 1st layer of SPE10. An average domain of 57.69% for pressure (top) and 8.18% for saturation (bottom) is solved over a course of several time-steps.

method are shown in Figure 4.13. Similar to the results for the previous case, lower the fluid compressibility, less local is the state evolution. On an average, 86.27% of the domain is flagged for the flow step and 23.38% for transport. Saturation update is invariably shaped by the underlying velocity field. In this case, due to the global update of pressure and a larger permeability contrast, support of the nonzero transport update is less local as well.



(a) Pressure results for the simulation of Case 2 with low oil compressibility.



(b) Saturation results for the simulation of Case 2 with low oil compressibility.

Figure 4.13: Localization results for the first layer of SPE10 case with different fluid properties and simulation controls.

The speedup obtained for pressure varies between 1.15 to 1.3 while in the case of saturation, we obtain 4.17 to 13.16 fold faster simulations. Figures 4.14a and 4.14b show the plots for the percentage of the domain flagged for the simulation of flow and transport with a maximum time-step size of 32 days. The average normalized run-times for each iteration are given in Table 4.9. The speedup obtained for pressure solve is between 1.5 to 1.88 fold and for saturation subsystem is between 5.42 to 26.04 fold.

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.83	0.78	0.76	0.135	0.072
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.15	1.18	1.30	4.17	13.16

Table 4.8: Comparison between localized and full Newton solver in terms of wall clock time for Case 2 (low compressibility).

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.63	0.55	0.52	0.08	0.034
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.50	1.63	1.88	5.42	26.04

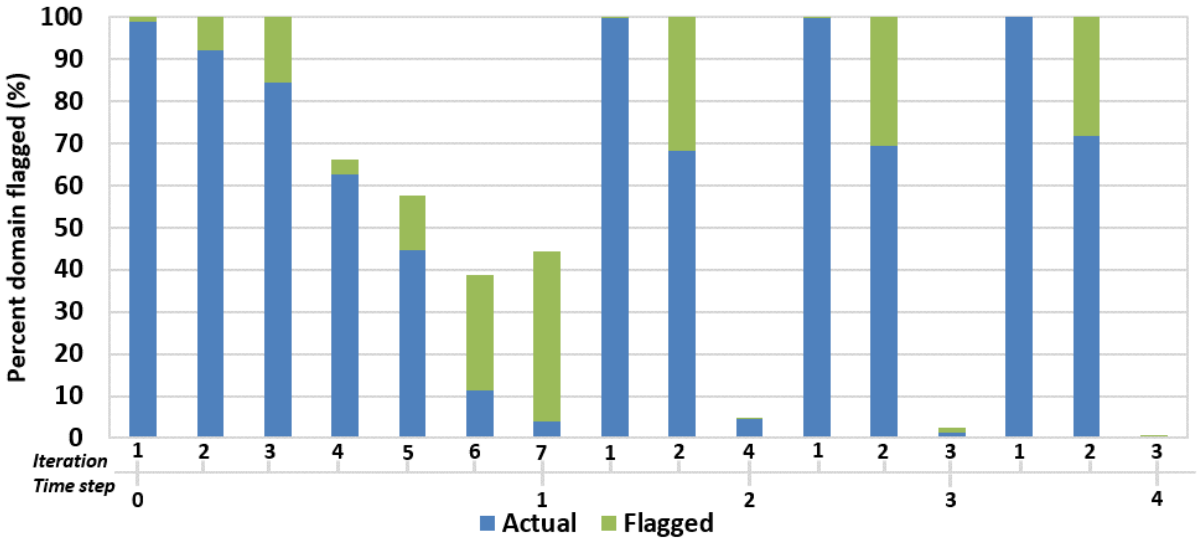
Table 4.9: Comparison between localized and full Newton solver in terms of wall clock time for Case 2 (big time-step).

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.57	0.48	0.44	0.0187	0.0048
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.64	1.86	2.21	8.14	113.63

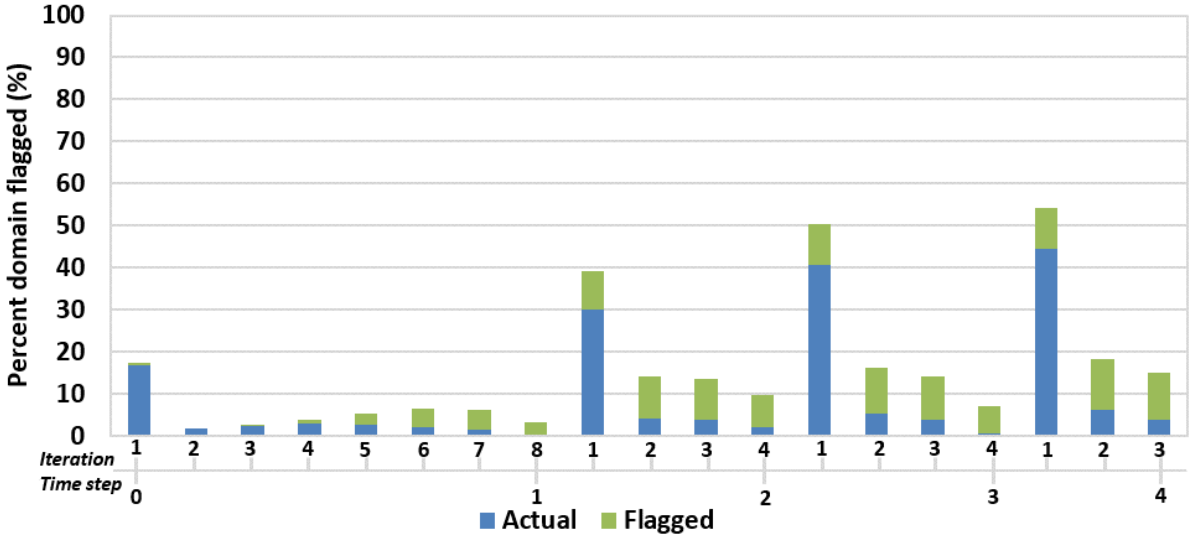
Table 4.10: Comparison between localized and full Newton solver in terms of wall clock time for Case 3 (high compressibility).

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.86	0.83	0.80	0.095	0.04
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.11	1.13	1.23	5.03	22.72

Table 4.11: Comparison between localized and full Newton solver in terms of wall clock time for Case 3 (low compressibility).



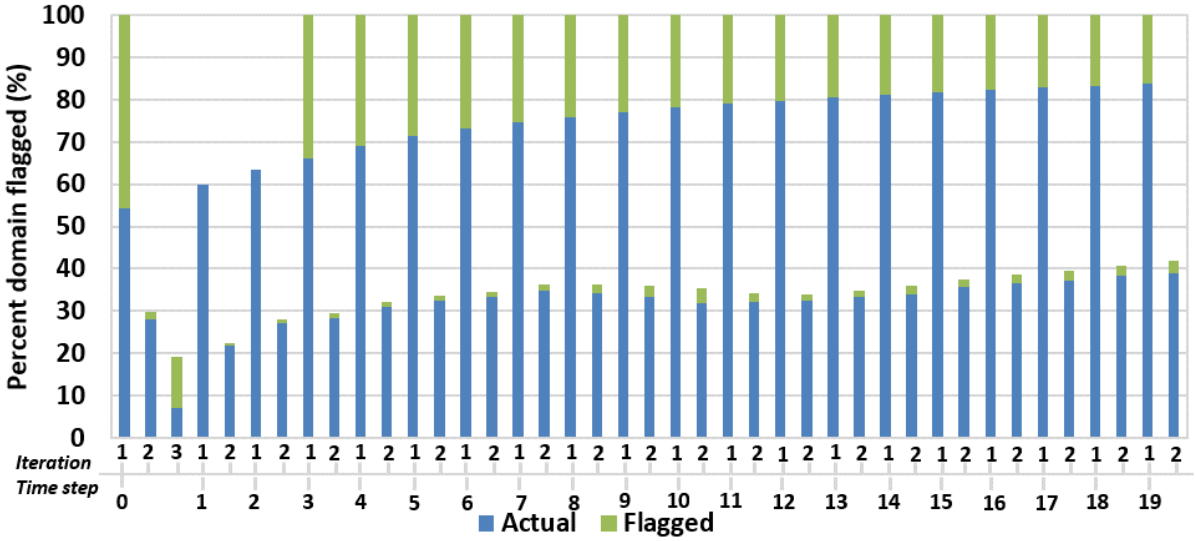
(a) Pressure results for the simulation of Case 2 with big time-steps.



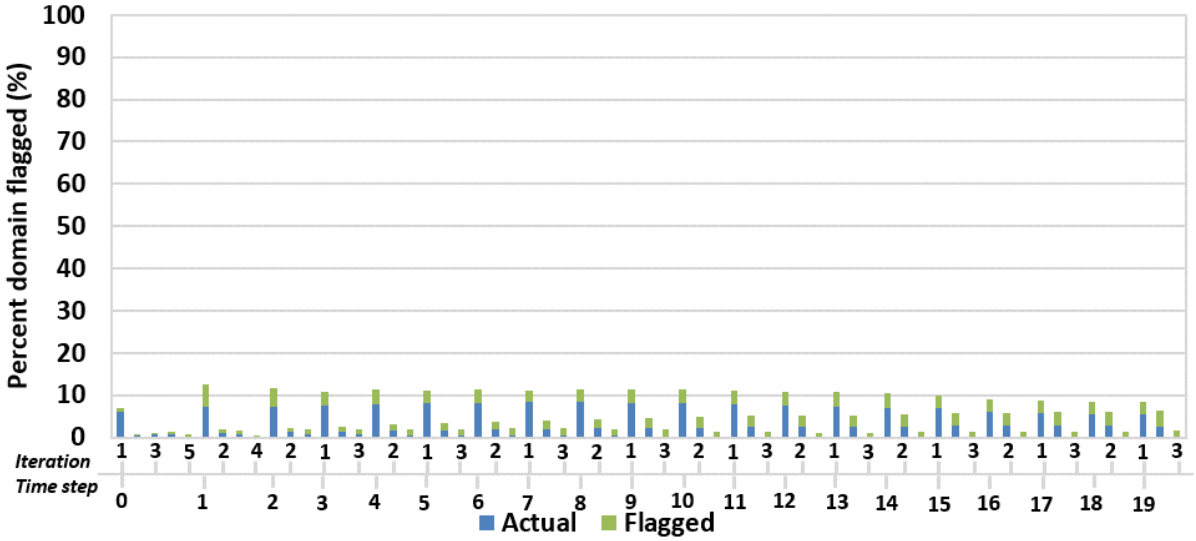
(b) Saturation results for the simulation of Case 2 with big time-steps.

Figure 4.14: Localization results for the first layer of SPE10 case with different fluid properties and simulation controls.

The final two-dimensional case is of a channelized permeability field with 2 producers and 1 injector (Table 4.3). Simulation of immiscible two-phase flow on the 48th layer of the SPE10 geological case with high oil compressibility results in Figure 4.15 and the flagging for this case can be seen in Figure 4.21 (third row). The trend here is similar to the ones observed in the previous cases with a slightly higher degree of locality. For the case with high oil compressibility, on an average, 64% and 5.22% of the domain is solved for pressure



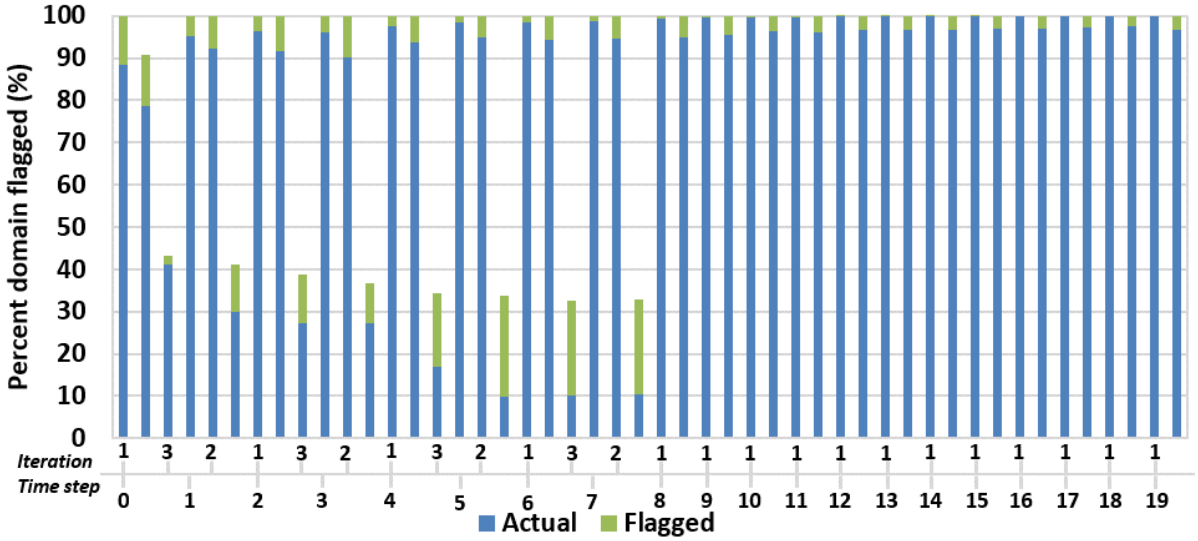
(a) Localization results for pressure evolution on the 48th layer of SPE10 case over the course of several time-steps.



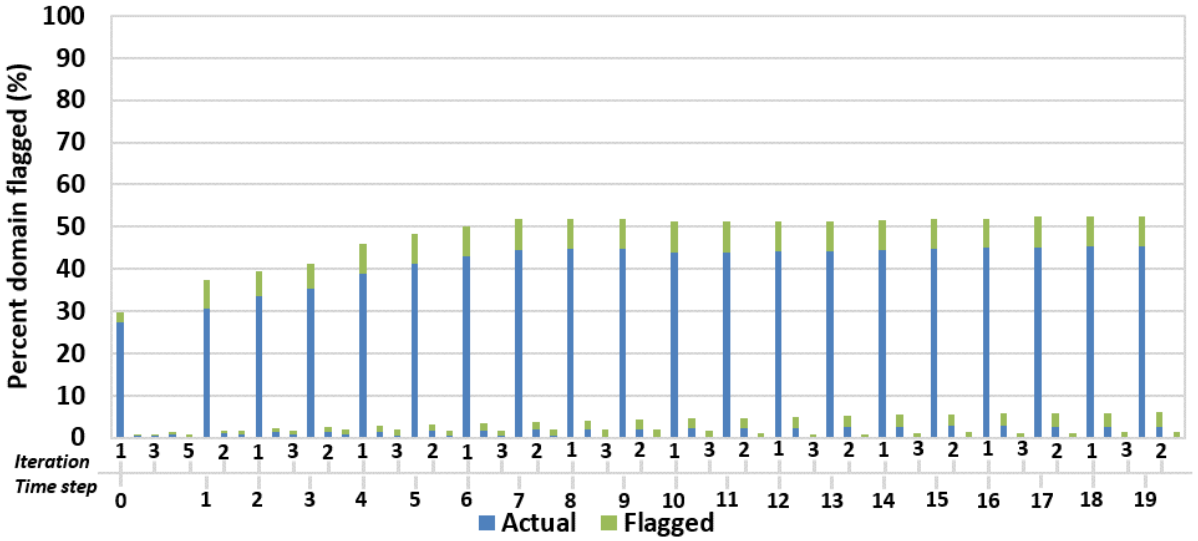
(b) Localization results for saturation evolution on the 48th layer of SPE10 case over the course of several time-steps.

Figure 4.15: Localization results for the case with high oil compressibility on the 48th layer of SPE10. An average domain of 64.19% for pressure (top) and 5.22% for saturation (bottom) is solved over a course of several time-steps.

and saturation, respectively. The computational speedups range from 1.64 to 2.21 fold for pressure and 8.14 to 113.63 fold for saturation subsystem. The cases with low oil compressibility and big time-step sizes show very promising results as well. We obtain a 5 to 22 fold improvement in computational time for transport solution with low oil compressibility.



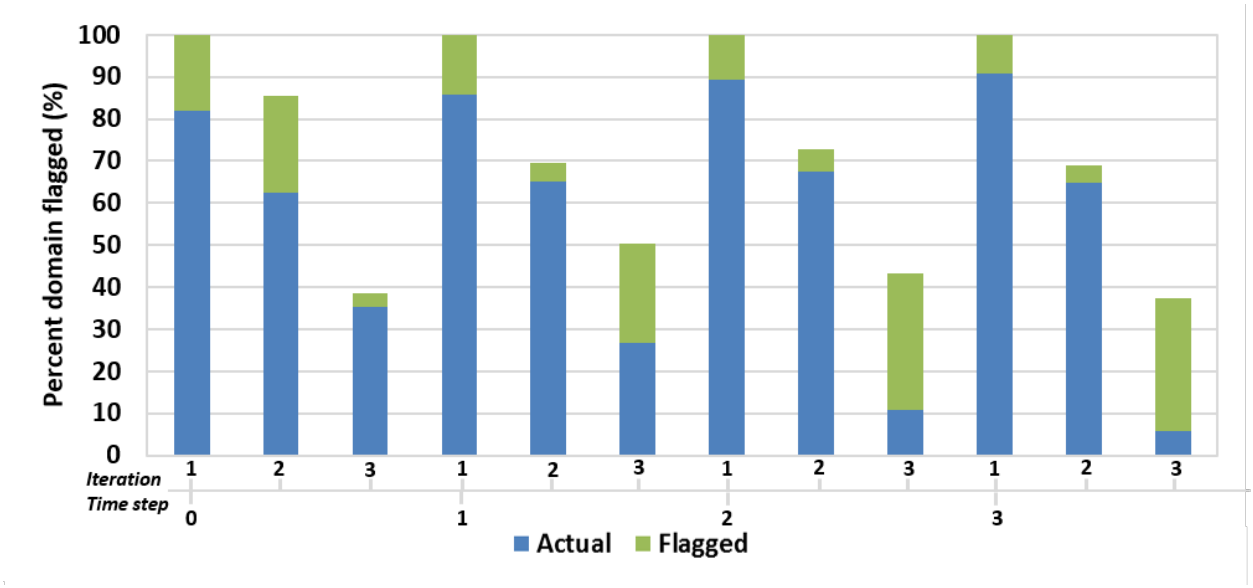
(a) Pressure results for the simulation of Case 3 with low oil compressibility.



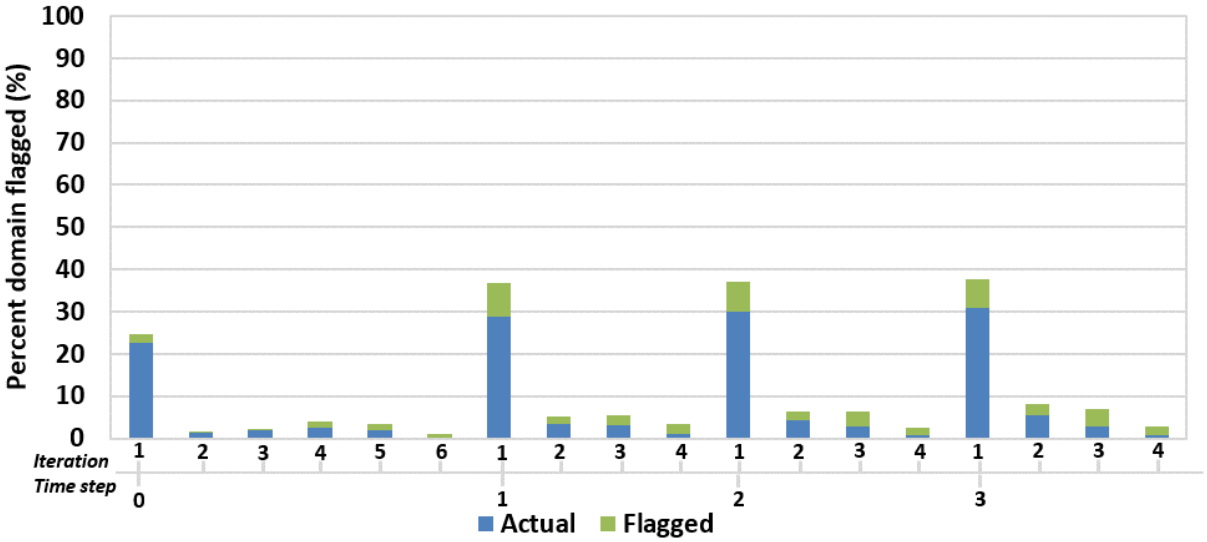
(b) Saturation results for the simulation of Case 3 with low oil compressibility.

Figure 4.16: Localization results for the 48th layer of SPE10 case with different fluid properties and simulation controls.

The gains are improved when the time-step sizes are big due to the increased number of iterations with a high degree of locality. We obtain around 7 fold faster simulations while using ILU(0)-GMRES as the preconditioner-linear solver combination. In the case of a direct solver, the improvement in computational time is drastic and can be as large as 48 folds.



(a) Pressure results for the simulation of Case 3 with big time-step size.



(b) Saturation results for the simulation of Case 3 with big time-step size.

Figure 4.17: Localization results for the 48th layer of SPE10 case with different fluid properties and simulation controls.

4.4 Effect of number of wells on locality

The effect of compressibility and time-step size has been discussed in the previous subsection. Along with these two factors, the number of wells also plays a major role in the degree of locality. Figure 4.18 shows the results for the percentage of domain exhibiting a nonzero (greater than 1.0E-06) Newton update. In this graph, blue bars represent the percentage of the domain updated for the case of two wells. As the number of wells increase,

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.66	0.59	0.56	0.047	0.017
Flagging	0.033	0.057	0.012	0.104	0.004
Speedup	1.44	1.53	1.75	6.62	48.08

Table 4.12: Comparison between localized and full Newton solver in terms of wall clock time for Case 3 (big time-step size).

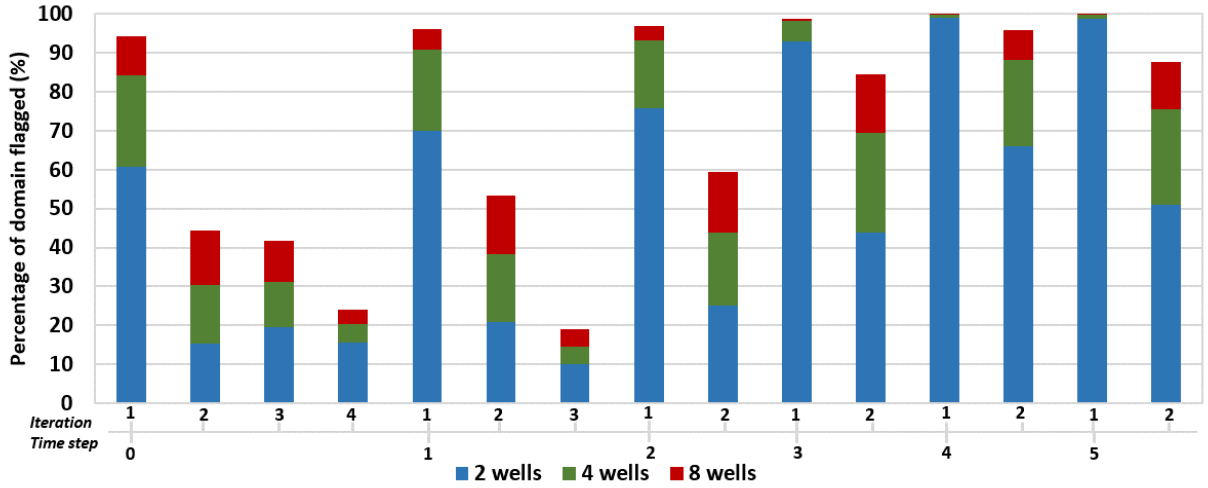
the initial locality decreases. After the first iteration, the degree of locality increases and even for the case of 8 wells, there is considerable locality for transport as well as for flow. Excessive number of wells in a small area will decrease the degree of locality considerably as the entire domain will experience a nonzero Newton update.

4.5 Three dimensional example

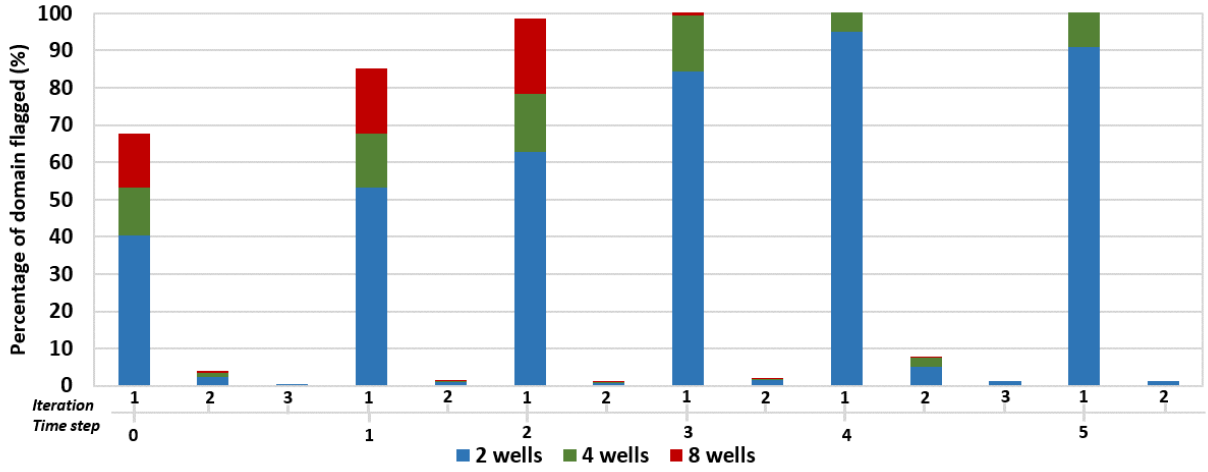
The final case is three dimensional SPE10 comparative study. There are two production wells at the diagonal vertices of the cuboid and one water injection well at the center of the reservoir. Figure 4.19a and 4.19b are obtained by the sequential simulation of flow and transport for 200 days. With an initial time-step size of 1 day, the biggest time-step used was 32 days. With big time-steps the pressure step exhibits a global update for the first iteration and then a decline in the area of the nonzero updates is observed. As it can be seen from Figure 4.19a, no visible trend can be observed and hence we emphasize on the need of a smart localized solver. For the case of saturation equation, updates are localized due to the hyperbolic nature of the governing equation. The computational cost is listed in Table 4.13, where the speedup obtained for the pressure equation varies between 1.87 to 2.72 times in computational time. For saturation, an increase of 5.62 to 16.43 folds is observed in the efficiency of the nonlinear solve.

4.6 Comparison with [36]

We compare the analytical method developed in this work with a heuristic method modeled after [36]. In this method the entire system is solved for the first iteration. The



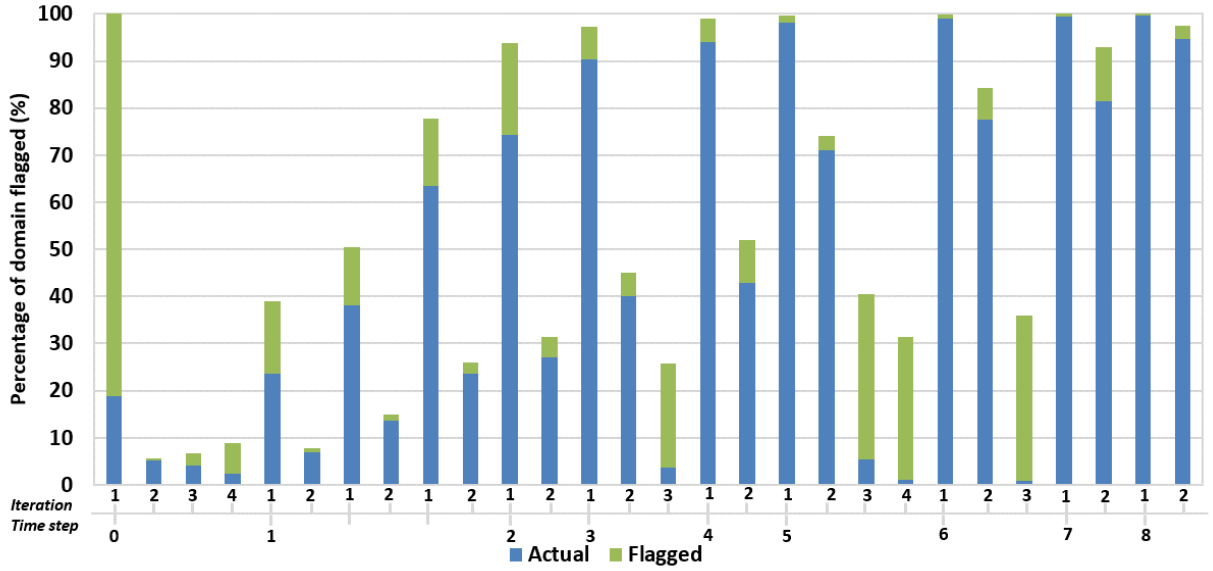
(a) Pressure solution results for 2, 4 and 8 wells on a heterogeneous permeability field.



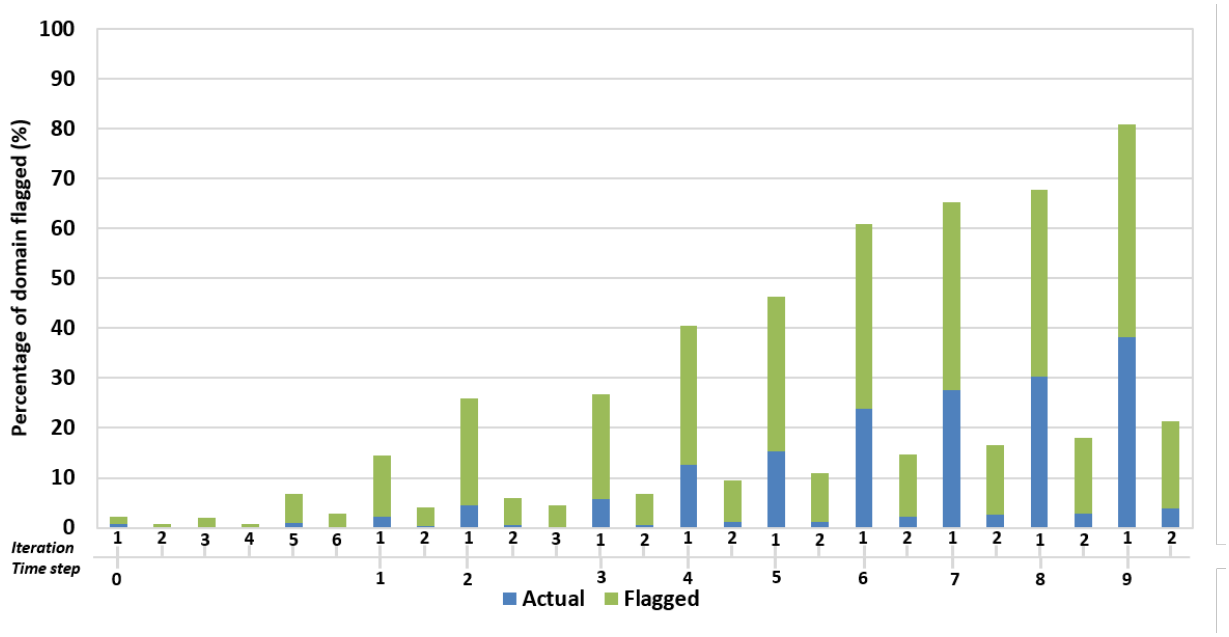
(b) Saturation solution results for 2, 4 and 8 wells on a heterogeneous permeability field.

Figure 4.18: Percentage of nonzero Newton updates for flow and transport with varying number of wells on the 48th layer of SPE10 comparative study case.

subsequent iterations are localized to the area of nonzero residual entries. As a safety measure, the support set is inflated using a tuning parameter. With the uncertainty in the support of the Newton update, heuristic methods might result in nonconservative estimates of the support set thereby degrading the nonlinear convergence rate. To illustrate the adverse effects of heuristic methods on the nonlinear convergence rate and the computational efficiency, we present the case with slight compressibility in the rock and fluids. With an initial time-step size of 10 days, we simulate flow and transport on the 48th layer of SPE10 comparative study for 100 days. In the case of pressure, the entire domain will be solved



(a) Pressure localization for a three dimensional case.



(b) Saturation localization for a three dimensional case.

Figure 4.19: Percentage of nonzero Newton updates for flow and transport for the three dimensional SPE 10 comparative study case.

at every time-step. In highly compressible flow, the pressure evolution extends outwards for the first iteration and then collapses towards the largest residual entry. In the case of low compressibility, the behavior of pressure evolution is unpredictable. Figure 4.20 is obtained by simulating the pressure equation over several time-steps and plotting the support of the

	Pressure step normalized time			Saturation step normalized time	
	AMG	ILU(0)	PARDISO	ILU(0)	PARDISO
Full Newton	1.0	1.0	1.0	1.0	1.0
Localized Newton	0.51	0.41	0.36	0.127	0.06
Flagging	0.023	0.011	0.000271	0.049	0.000148
Speedup	1.87	2.36	2.72	5.62	16.43

Table 4.13: Comparison between localized and full Newton solver in terms of wall clock time for a three dimensional SPE10 case.

nonzero Newton updates. Due to the inability of the heuristic method to predict locality conservatively, it took 61 iterations for the entire simulation (with an average of a seven-fold degradation in convergence). As a precaution we have inflated the support set by 1 extra layer. Using the proposed solver, the same simulation took 8 iterations. Also, solving the entire problem using traditional Newton's method, it took the same number of iterations as the localized Newton solver.

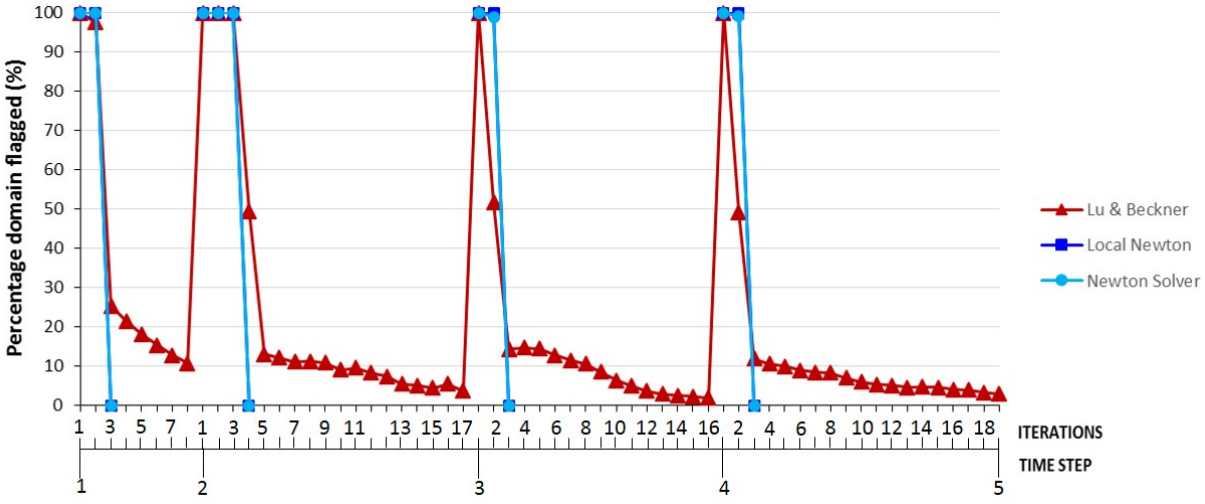


Figure 4.20: Comparison of the proposed analytical method with a heuristic approach to predict locality for an incompressible flow problem. The nonlinear convergence is severely affected by using the heuristic approach, thereby degrading the computational efficiency.

The complexity analysis for both of these methods is provided in Table 4.14. The speedup is calculated using the normalized time. The speedup obtained for the analytical localization algorithm is 1.0 because the area flagged by the analytical estimate is the entire domain, which is the case for the numerical update. As described in Algorithm 3, if the active

	ILU(0)-GMRES	PARDISO
Heuristic pressure	0.72	0.93
Analytical pressure	1.0	1.0
Heuristic saturation	2.12	2.27
Analytical saturation	4.10	5.77

Table 4.14: Comparison between the speedup (in run time) obtained for the localized solver and the heuristic method with respect to the full Newton solver.

set is greater than M% (Algorithm 2) of the domain, the flagging step is skipped and the time taken would be negligible. Hence the time taken to solve the linear system would be equal to the time required to solve the full matrix. On the other hand, due to the inability of the heuristic method to predict the nonzero support of the Newton update conservatively, there is a deterioration of the computational speed. For the case of saturation, an average speedup obtained for Lu & Beckner's method is 2.2 compared to 5.0 for the localized solver. We can easily see that the heuristic method failed to capture the uncertainty in the support of the nonzero Newton updates, thereby affecting the nonlinear convergence and the computational efficiency of the solver.

4.7 Application to three phase flow problem

The canonical forms of the three-phase flow equations are derived in Appendix B. The *quasilinearization* operator applied to Equations B.0.9, B.0.10 and B.0.11 and the subsequent decoupling of the flow equation from the transport will result in

$$\begin{bmatrix} F_1^1 & F_1^2 & F_1^3 \\ F_2^1 & F_2^2 & F_2^3 \\ F_3^1 & F_3^2 & F_3^3 \end{bmatrix} \begin{bmatrix} \delta_1^f \\ \delta_2^f \\ \delta_3^f \end{bmatrix} = - \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \qquad \Longrightarrow \qquad \begin{bmatrix} F_1^1 & 0 & 0 \\ 0 & F_2^2 & 0 \\ 0 & F_3^2 & F_3^3 \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = - \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{4.7.1}$$

where F_m^p, $m, p \in \{1, 2, 3\}$, is the derivative of the m^{th} equation with respect to the p^{th} variable, δ^f is the Newton update for the fully coupled system and δ is the infinite dimensional Newton update for the sequential simulation of flow and transport. This formulation assumes that gas does not dissolve in water and hence the water governing equation is automatically

decoupled from the gas mass balance equation. Given this reduction, we can apply the same formulas developed for the two phase flow case and apply them to three different equations in this case as opposed to just two scalar equations. Problem specific details are given in [49].

4.7.1 Locality

In this section the localization algorithm is implemented in a three-phase flow simulator for different problem settings. Solutions for the radius of effect given by Equations 3.4.27 and 3.5.24 are used to obtain the flagged domain that needs to be solved at each Newton step. Flagging results for the first layer in the SPE10 comparative study case are shown in Figure 4.21. In this figure the left column shows the pressure localization, middle column shows the localization for water saturation and the right column gives the gas saturation results. In all the figures below, blue shaded region depicts the actual support of the nonzero (greater than the tolerance) Newton's update which is obtained by the numerical simulation of a full nonlinear step. The green shaded region represents the area that is flagged by the analytical algorithm developed in the previous chapters.

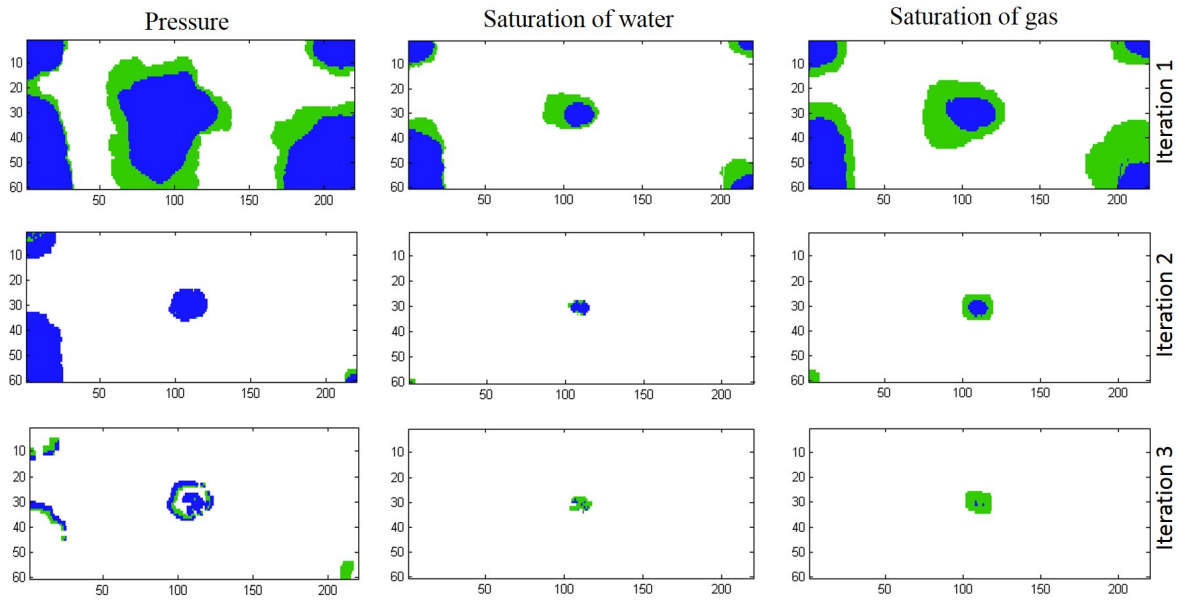
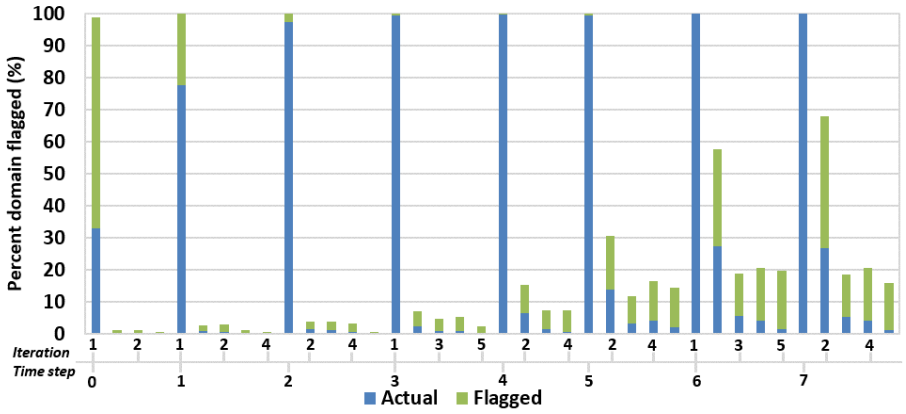
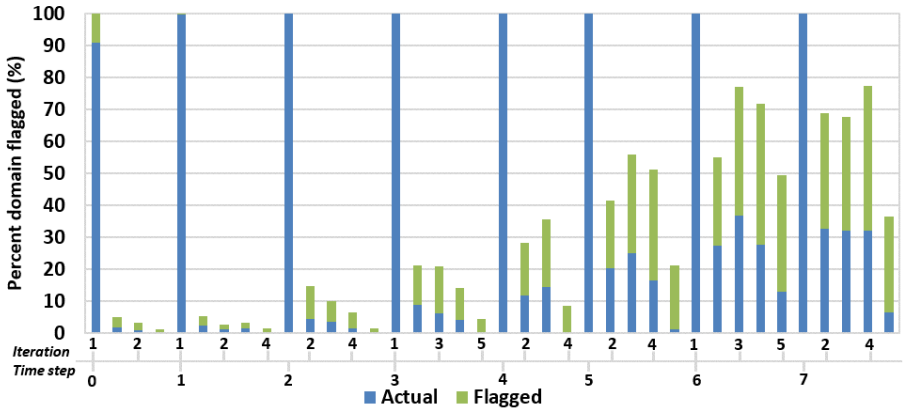


Figure 4.21: Flagging results for pressure, S_w and S_g for the first layer of SPE10. Results for the other cases show similar conservativeness.

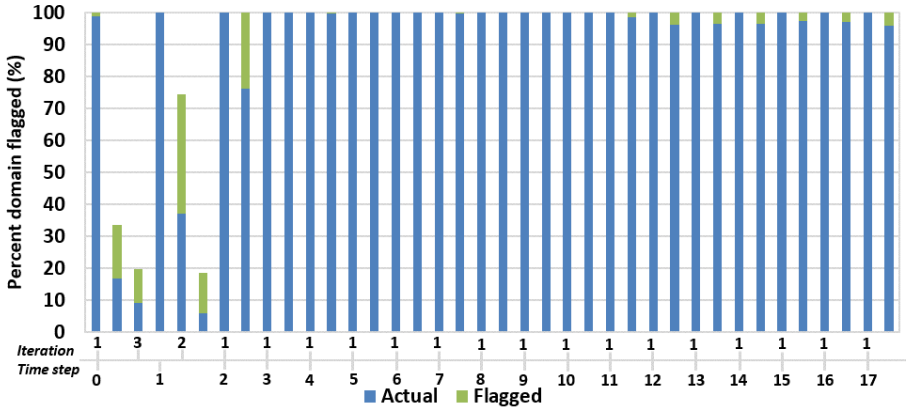
Simulation results are provided for each of the cases described in the previous section



(a) Localization results for saturation of water over the course of several time-steps.

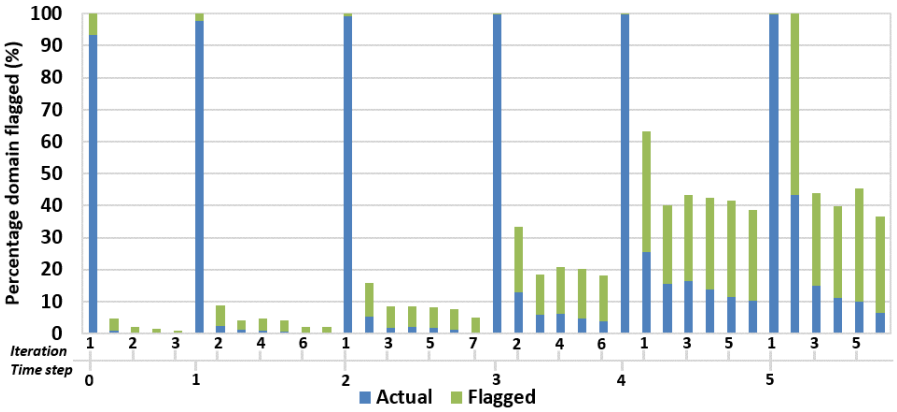


(b) Localization results for gas saturation over the course of several time-steps.

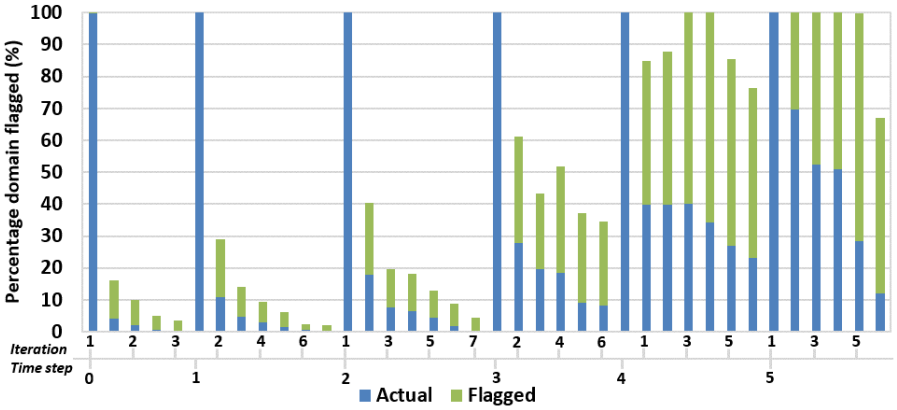


(c) Localization results for pressure of oil over the course of several time-steps.

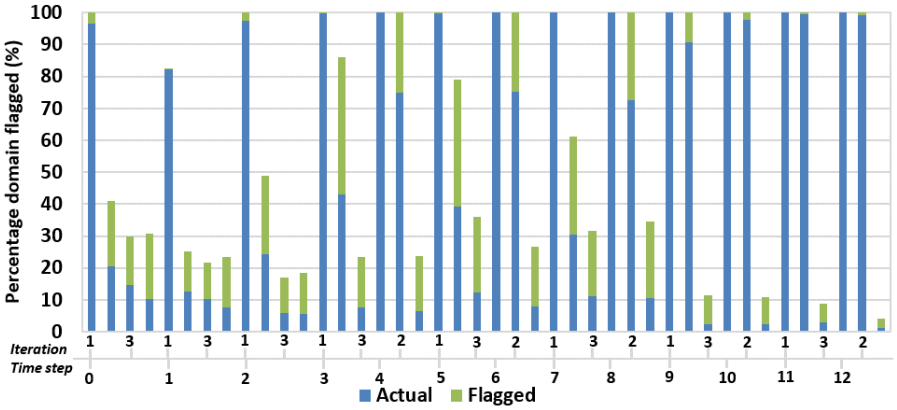
Figure 4.22: Localization results for the case with homogeneous permeability field over several time-steps. An average domain of 88.95% for pressure (bottom), 27.17% for S_w (top) and 34.84% for S_g (middle) is solved for this particular simulation case.



(a) Localization results for water saturation over the course of several time-steps.

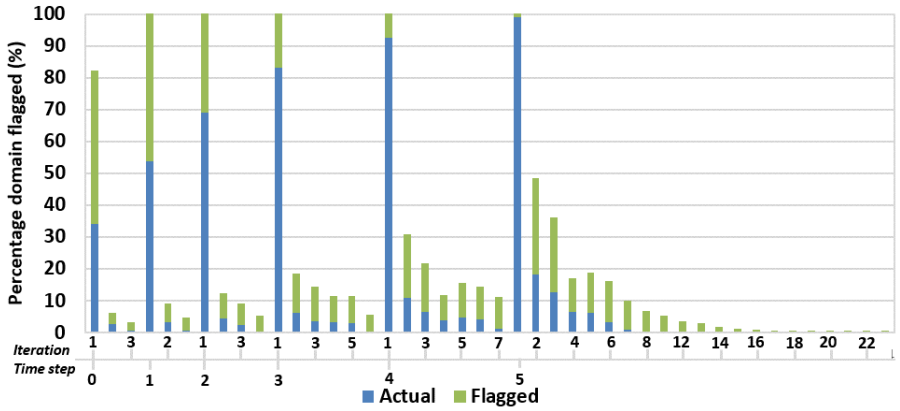


(b) Localization results for gas saturation over the course of several time-steps.

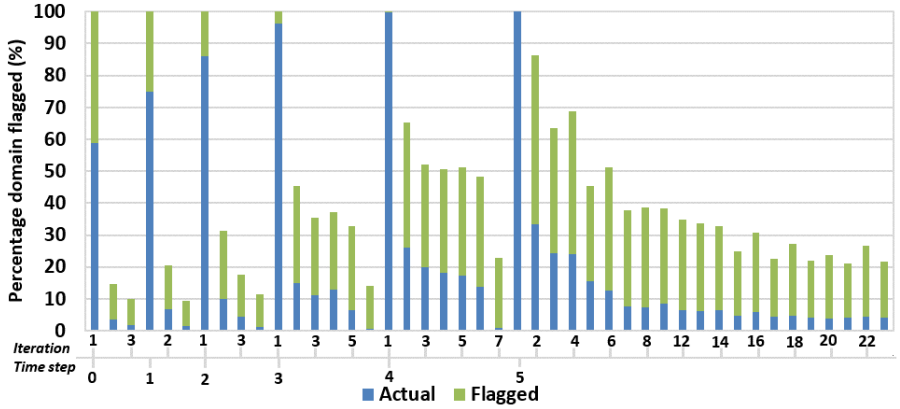


(c) Localization results for pressure of oil over the course of several time-steps.

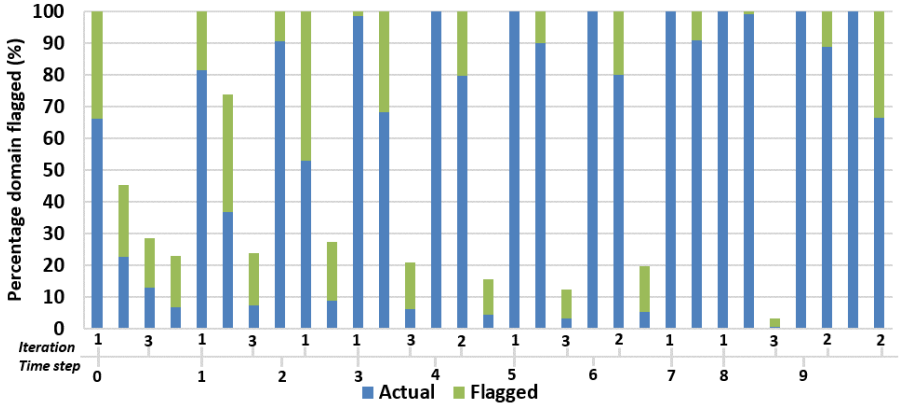
Figure 4.23: Localization results for the case with Gaussian permeability field. An average domain of 70.59% for pressure (bottom), 26.93% for S_w (top) and 40.15% for S_g (middle) is solved over a course of several time-steps.



(a) Localization results for water saturation over the course of several time-steps.



(b) Localization results for gas saturation over the course of several time-steps.



(c) Localization results for pressure over the course of several time-steps.

Figure 4.24: Localization results for the case with channelized permeability field. An average domain of 80.46% for pressure (bottom), 18.87% for S_w (top) and 26.36% for S_g (middle) is solved over a course of several time-steps.

along with the three-phase properties listed in Appendix D. The initial time-step size is 0.5 days and the maximum is 35.0 days with a growth factor of 2. The simulations are ran for 600 days and for comparison purposes, final results are provided as well. Better results can be obtained for compressible fluids and less well density. In all the localization results below, the plots shown are only for a few time-steps due to a huge amount of data. The full simulation results are analogous to the ones shown in the figures. The speedup in the computational time are calculated for the entire simulation run unlike the localization plots. Figure 4.22 is obtained by the simulation of Case 1 with varying time-step sizes. Higher fluid compressibility results in smaller pressure diffusion and thus higher degree of locality in the solution process. In Figures 4.22, 4.23 and 4.24, blue bars represent actual percentage of domain solved by the simulator while the green bars are the percentage of domain that is flagged by the proposed method. The x-axis shows the number of time-steps and the number of iterations in each step. From the figure it can be inferred that the support of significant Newton updates is large. In the case of homogeneous permeability field, the resulting nonlinear system is not very complex and thus require smaller number of Newton iterations to converge. Notice that the first Newton update in each time step for the pressure solution is almost always global. If the Newton process takes fewer iterations to converge the average domain updated over the course of simulation will be higher. In the case of problems with higher complexity, the required number of Newton iterations increase and thus as the iterations proceed the locality grows bigger. Hence, the average domain solved over the course of the simulation is smaller. Similar conclusions can be drawn for the time-step size. Several experiments were performed with different time-step sizes. It was observed in some cases that bigger time-steps require more number of Newton iterations and thus the degree of locality in such problems is higher. Figure 4.23 shows the localization results for the first layer of SPE10 where the permeability field follows the Gaussian distribution. This problem is slightly more difficult to solve than the homogeneous and have a higher variation in the local wave speeds due to the underlying heterogeneous permeability field. This increases the locality in the underlying physics and thus increase the efficiency of the localized Newton

	Pressure step time (sec)			Saturation step time (sec)		
	AMG	ILU(0)	PARDISO	AMG	ILU(0)	PARDISO
Full Newton	1.5E-02	5.2E-03	2.5E-02	-	5.0E-03	7.2E-02
Localized Newton	1.2E-02	4.2E-03	2.2E-02	-	1.7E-03	2.0E-02
Flagging	5.0E-04	5.0E-04	5.0E-04	1.0E-03	1.0E-03	1.0E-03
Speedup	1.17	1.11	1.19	-	1.9	3.09

Table 4.15: Complexity analysis for localized linear solver for the two-dimensional homogeneous case (60X220 gridcells).

	Pressure step time (sec)			Saturation step time (sec)		
	AMG	ILU(0)	PARDISO	AMG	ILU(0)	PARDISO
Full Newton	1.5E-02	5.2E-03	2.7E-02	-	5.0E-03	7.2E-02
Localized Newton	1.1E-02	2.9E-03	1.4E-02	-	1.2E-03	8.6E-03
Flagging	5.0E-04	5.0E-04	5.0E-04	1.0E-03	1.0E-03	1.0E-03
Speedup	1.57	1.52	1.72	-	2.57	6.48

Table 4.16: Complexity analysis for localized linear solver for the first layer of SPE10 (60X220 gridcells).

solver. For the case of the forty-eighth layer of SPE10 (Figure 4.24), we can see that pressure evolution is less global over the course of the simulation which points to higher degree of locality.

In the Tables 4.15, 4.16 and 4.17, we present the simulation run times and the increase in the computational time for the localized Newton solver. In all the tables, times reported are the wall clock timings for each solve. For the full Newton solve, we directly use the time from the simulator and then use the complexity curves developed in Figure 3.14 to calculate the timings for the local solve. Table 4.15 gives the timings for the homogeneous permeability

	Pressure step time (sec)			Saturation step time (sec)		
	AMG	ILU(0)	PARDISO	AMG	ILU(0)	PARDISO
Full Newton	1.5E-02	5.2E-03	2.7E-02	-	5.0E-03	7.2E-02
Localized Newton	9.6E-03	3.0E-03	1.8E-02	-	5.7E-04	5.4E-03
Flagging	1.5E-03	1.5E-03	1.5E-03	7.5E-04	7.5E-04	7.5E-04
Speedup	1.36	1.3	1.4	-	4.0	12.63

Table 4.17: Complexity analysis for localized linear solver for the forty eighth layer of SPE10 (60X220 gridcells).

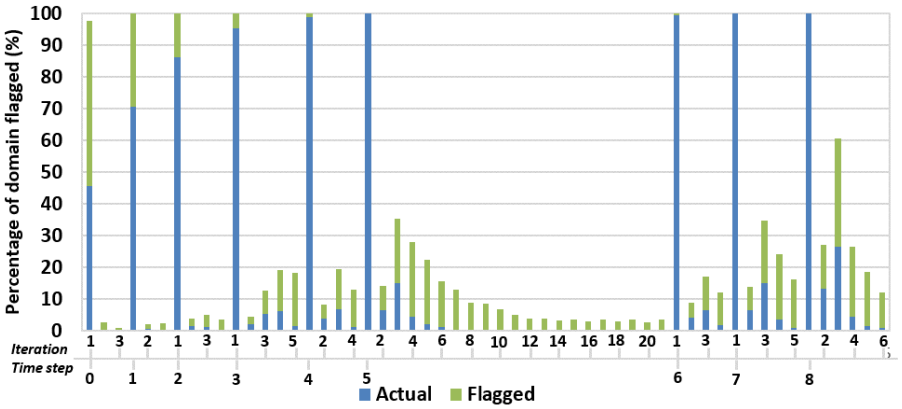
	Pressure step time (sec)			Saturation step time (sec)		
	AMG	ILU(0)	PARDISO	AMG	ILU(0)	PARDISO
Full Newton	0.25	0.67	1.03	-	0.44	3.45
Localized Newton	0.17	0.4	0.61	-	0.05	0.31
Flagging	0.051	0.051	0.051	0.05	0.05	0.05
Speedup	1.13	1.5	1.56	-	4.2	9.6

Table 4.18: Complexity analysis for localized linear solver for the three dimensional case (60X220X10 gridcells).

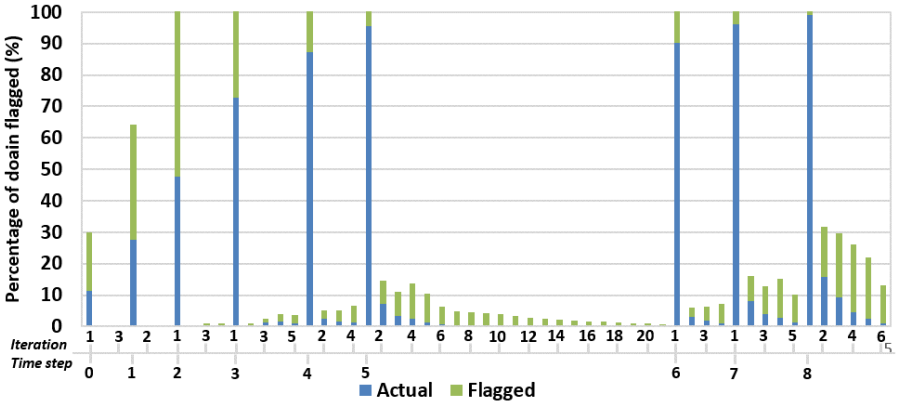
field case wherein the maximum speedup obtained is about 3 times for saturation due to less local evolution of the Newton updates. In case of pressure, there is just marginal improvement of around 20%. While in the case of the SPE10 Gaussian field, pressure solve is almost twice as fast as the full Newton's method and the saturation solve is around 7 folds faster for PARDISO direct solver. The best performance gains are obtained in the case of the channelized permeability layer due to the large contrasts in the local wave speeds. Transport solve in this case is between 4 to 13 times faster. Overall, the speedups can be improved in the cases where there are fewer wells or variable well controls. Also, localized solver will perform better in cases with large oil compressibility.

4.7.2 Three dimensional example (SPE10 - $60 \times 220 \times 10$)

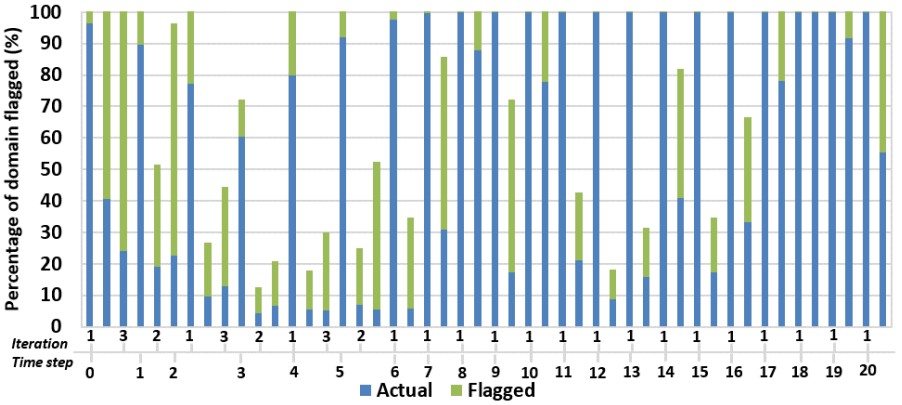
The next test example is a three-dimensional problem with the first 10 layers of the SPE10 comparative case. The layers show a Gaussian permeability distribution and a heterogeneous porosity field. Please refer to Appendix D for detailed problem description. Table 4.18 shows the computational gains obtained in a three dimensional case. For coupled saturation the performance improvement is around 4 to 10 folds and for pressure, due to a more global evolution, the solution is around 50% faster. The estimates derived in the theory sections are for general three-dimensional problems. The same estimates work for the two-dimensional cases. The localization results are shown in Figure 4.25. Experiments show that the method is conservative for problems with gravity as well, though the extent of conservativeness needs further investigation.



(a) Localization results for water saturation over the course of several time-steps.



(b) Localization results for gas saturation over the course of several time-steps.



(c) Localization results for pressure over the course of several time-steps.

Figure 4.25: Localization results for the three dimensional SPE10 case. An average domain of 74.74% for pressure (bottom), 21.35% for S_w (top) and 31.12% for S_g (middle) is solved over a course of several time-steps.

CHAPTER 5

EXTENSION TO FULLY IMPLICIT MULTI-COMPONENT SCHEME

The mathematical framework proposed in the previous chapters is formulated based on the sequential implicit schemes. This entails the solution of decoupled flow and transport equations as described in Algorithm 4. This scheme allows the use of specialized solvers to exploit the specific characteristics of the elliptic flow and hyperbolic transport equations. An often cited disadvantage of the sequential schemes is that it degrades the Newton convergence rate for a tightly coupled system. A widely used alternative is the fully coupled implicit scheme that solves the flow and transport equations simultaneously, which has better convergence properties as opposed to the sequential schemes. In this chapter a heuristic method is proposed which uses the framework developed for the sequential model and applies it to a fully coupled case. Certain well-known preconditioners are used to improve the Newton convergence rate for the problems tested. Even though the examples presented in this chapter cover a wide range of problems, the reader is advised to take caution while implementing such frameworks, as it being heuristic in nature, previous claims might not hold true in every possible permutation and combination of heterogeneity and complexity. Further extensive testing is required to develop proofs and claims of such nature. Developing strong theoretical justifications fall under the scope of future work. Along with extension to fully coupled case, systems of hyperbolic equations are solved instead of scalar transport equations.

5.1 Conservation equations and constraints

The transport equations for a system containing n_c components and n_p phases can

be written as

$$V_b \frac{\partial}{\partial t}\left[\phi\left(\sum_p \rho_p S_p\right) z_c\right] = -\nabla \cdot \left[\left(\sum_p \frac{x_{c,p}\rho_p k_{rp}}{\mu_p}\mathbf{K}\left(\nabla P + \bar{\gamma}_p \nabla D\right)\right)\right] + \sum_p q_{c,p}, \quad (5.1.1)$$

where $p = 1, ..., n_p$, $c = 1, ..., n_c$, z_c is the overall molar fraction of species c, $x_{c,p}$ is the mole fraction of component c in phase p, P is the phase pressure, $\bar{\gamma}_p$ is the phase mass density and $q_{c,p}$ is the source/sink term for each species present in a particular phase, p. Mass transfer is allowed only between oil and gas phases and water component is present only in the water phase while the hydrocarbon components are present in the non-aqueous phases. In this study, capillary pressure is assumed to be zero and the temperature to be constant. To close the system, additional equations are required. These include the equations for thermodynamic equilibrium, given by

$$f_{c,p}(P, x_{c,p}) - f_{c,q}(P, x_{c,q}) = 0, \quad p \neq q, \quad c = 1, ..., n_c, \quad (5.1.2)$$

where $f_{c,p}$ is the fugacity of component c in phase p. The overall compositions and phase saturations sum to unity, given by

$$\sum_{c=1}^{n_c} z_c - 1 = 0, \text{ and } \sum_{p=1}^{n_p} S_p - 1 = 0. \quad (5.1.3)$$

The above equations provide a complete mathematical statement for isothermal multiphase multi-component flow in porous media.

Overall composition variable formulation is used in this research with the primary variables

1. Oil phase pressure, P, aligned with the total mass conservation equation,

$$V_b \frac{\partial}{\partial t}\left[\phi\left(\sum_p \rho_p S_p\right)\right] = -\nabla \cdot \left[\left(\sum_p \frac{\rho_p k_{rp}}{\mu_p}\mathbf{K}\left(\nabla P + \bar{\gamma}_p \nabla D\right)\right)\right] + \sum_p q_p, \quad (5.1.4)$$

where q_p is the source/sink term for phase p.

2. Water saturation, S_w, aligned with the water conservation equation,

$$V_b \frac{\partial}{\partial t}\left[\phi\left(\rho_w S_w\right)\right] = -\nabla \cdot \left[\left(\frac{\rho_w k_{rw}}{\mu_w}\mathbf{K}\left(\nabla P + \bar{\gamma}_w \nabla D\right)\right)\right] + q_w \tag{5.1.5}$$

3. $n_c - 1$ overall molar fractions, Z_c, aligned with $n_c - 1$ component conservation equations,

$$V_b \frac{\partial}{\partial t}\left[\phi\left(\sum_p \rho_p S_p\right) z_c\right] = -\nabla \cdot \left[\left(\sum_p \frac{x_{c,p}\rho_p k_{rp}}{\mu_p}\mathbf{K}\left(\nabla P + \bar{\gamma}_p \nabla D\right)\right)\right] + \sum_p q_{c,p} \tag{5.1.6}$$

In this model, an instantaneous thermodynamic equilibrium is assumed for a given Newton iteration. For each iteration, a phase stability test is performed to determine if the overall composition will split into two phases. If a two-phase state is expected, flash calculations are performed to obtain the phase compositions, $x_{c,p}$. Thereafter, the fugacity constraint is satisfied for a given thermodynamic state. The advantage of using an overall composition formulation is that the equation and variable set remain constant for the entire grid and are well defined, which removes the need for variable substitution step. However, phase equilibrium calculations need to be performed for every discrete element which might result in increased Equation of state (EoS) solution costs.

5.2 Heuristic extension of the mathematical framework

The key objective is the development of an algorithmic process for the identification of the elements of a Newton update that may be neglected, prior to solving the linear hyperbolic system given by Equations 5.1.5 and 5.1.6. The size of the linear system for the hyperbolic part is reduced to match the size of the domain resulting in a significant change in the state variables. Equation 5.1.4 which is near elliptic in nature, will be solved on the entire grid, similar to fully implicit Newton step, due to a more global evolution of the pressure variable.

5.2.1 *Canonical hyperbolic system*

Equations 5.1.5 and 5.1.6 can be generalized by

$$\bar{R}_\infty(z_c, S_w; P) = \frac{\partial}{\partial t} a(z_c, S_w; P) + \nabla \cdot [\mathbf{b}(z_c, S_w; P) + \mathbf{g}(z_c, S_w; P)] - w(z_c, S_w; P), \quad (5.2.1)$$

where $\bar{R}_\infty$ is the nonlinear residual operator,

$$a = \begin{cases} V_b \phi \rho_p S_p, & p = w \\ V_b \phi \left(\sum_p \rho_p S_p \right) z_c, & p = o, g, \end{cases} \quad (5.2.2)$$

$$\mathbf{b} = \begin{cases} \frac{\rho_p k_{rp}}{\mu_p} \mathbf{K} \nabla P, & p = w \\ \sum_p \frac{x_{c,p} \rho_p k_{rp}}{\mu_p} \mathbf{K} \nabla P, & p = o, g, \end{cases} \quad (5.2.3)$$

$$\mathbf{g} = \begin{cases} \frac{\rho_p k_{rp}}{\mu_p} \mathbf{K} \bar{\gamma}_p \nabla D, & p = w \\ \sum_p \frac{x_{c,p} \rho_p k_{rp}}{\mu_p} \mathbf{K} \bar{\gamma}_p \nabla D, & p = o, g, \end{cases} \quad (5.2.4)$$

and

$$w = \begin{cases} q_p, & p = w \\ \sum_p q_{c,p}, & p = o, g. \end{cases} \quad (5.2.5)$$

Discretizing Equation 5.2.1 in time and leaving the space variables continuous, we obtain

$$R_\infty(z_c, S_w; P) = a(z_c, S_w; P) - a^n(z_c, S_w; P) + \Delta t \nabla \cdot [\mathbf{b}(z_c, S_w; P) + \mathbf{g}(z_c, S_w; P)] - \Delta t w(z_c, S_w; P), \quad (5.2.6)$$

where all the terms are evaluated implicitly, i.e. at time-step $(n+1)$, while a^n represents the explicit term in the temporal discretizaiton.

5.2.2 *Infinite dimensional Newton iteration*

Assuming Fréchet differentiability (denoted by R'_∞), and invertibility of the derivative, Newton's method can be defined in infinite dimensions. In the infinite form, Newton's

method may be applied to solve Equation 5.2.6. As the state variable field, $u(x) : \Omega \to \mathbb{R}^N$, remain fixed for one Newton iteration, we can write the functional dependence of each coefficient in Equation 5.2.6 in terms of space alone. Fréchet derivative of Equation 5.2.6 can be derived as

$$R'_\infty(x)\delta^\nu_\infty = A(x)\delta_\infty(x) + \nabla \cdot [\mathbf{B}(x)\delta_\infty(x)]. \tag{5.2.7}$$

Given the number of equations, N_{eq}, and the number of variables, N_{var}, with $i = 1, ..., N_{eq}$ and $j = 1, ..., N_{var}$, coefficient matrix $A_{ij} = \frac{\partial(a_i - \Delta t w_i)}{\partial u_j}$ and $B_{ij} = \Delta t \frac{\partial(\mathbf{b}_i + \mathbf{g}_i)}{\partial u_j}$, where $u = [S_w, z_1, ..., z_{n_c-1}]^T$. Subsequently, given a Banach space Ω and the domain boundary $\partial\Omega$, the infinite dimensional Newton iteration can be written as

$$\nabla \cdot [\mathbf{B}(x)\delta_\infty(x)] + A(x)\delta_\infty(x) + R_\infty(x) = 0 \quad x \in \Omega, \tag{5.2.8}$$

with boundary condition

$$\delta_\infty(x) = 0 \quad x \in \partial\Omega. \tag{5.2.9}$$

We seek closed form solution of Equation 5.2.8 and 5.2.9 to obtain $\delta(x)$. Due to the variable nature of the coefficient matrices, $\mathbf{B}$ and A, analytical solutions are seldom tractable. In order to estimate the solution of the above equations, we introduce two simplifying assumptions similar to the ones developed in this work that guarantee conservative estimates. Reiterating, the simplifications are as follows:

1. Homogenization of variable coefficient matrices that contain permeability field, porosity field, gravity and well terms.

2. Assuming radially symmetric solutions.

The simplified equation thus becomes

$$\begin{cases} B_{max}\frac{d}{dr}[\delta^*_\infty(r)] + A_{min}\delta^*_\infty(r) = -R_\infty(r), & r \in \Omega, \\ \delta^*_\infty(r) = 0, & r \in \partial\Omega, \end{cases} \tag{5.2.10}$$

where, δ^*_∞ is the radially symmetric Newton's update obtained on the homogenized domain. Ω.

5.2.3 Solution to Equation 5.2.10

Equation 5.2.10 is linear and one dimensional. Superposition may be applied by decomposing the projected residual into a sum of local bump functions that are piecewise nonzero over each control volume in the grid. That is, let V_i denote one control volume in the domain. Suppose that $R_{h,i}$ is the component of the discrete residual corresponding to the control volume obtained from the simulator. Then,

$$R_\infty(r_{h,i}) = [I_h^\infty R_h]_i := R_{h,i} H_{V_i},$$

where H_{V_i} is the Heaviside distribution. Therefore, for each element, we can shift the origin to the centroid of the control volume and assume that the control volume can be represented by a sphere. Equation 5.2.10 can be transformed into

$$\begin{cases} B_{max} \frac{d}{dr}[\delta^*_\infty(r)] + A_{min}\delta^*_\infty(r) = 0, & r \geq r_{h,i}, \\ \delta^*_\infty(r) = \delta^*_{h,i}, & r \leq r_{h,i}, \end{cases} \tag{5.2.11}$$

where $\delta^*_{h,i}$ is obtained from the simulator by solving the block diagonal Jacobian matrix and neglecting the derivatives on the off diagonal blocks. Consequently, the solution of Equation 5.2.11 is given by

$$\delta^*_\infty(r) = \exp\left[-{B_{max}}^{-1} * A_{min} * (r - r_{h,i})\right]\delta^*_{h,i}. \tag{5.2.12}$$

For a given $\left|\delta^*_{h,i}\right| > \delta_\epsilon$, where δ_ϵ is the cutoff value of the Newton update, it is observed that the Newton updates, $|\delta^*_\infty(r)|$, damp monotonically with increasing r. Figure 5.1 shows the behavior of $\delta^*_\infty(r)$ along the grid for a nonzero forcing term at $i = 250$. This observation prompts an alternative algorithm wherein instead of computing the Newton update for each

control volume V_i for every $\left|\delta^*_{h,i}\right| > \delta_\epsilon$ and then using superposition, radius r can be computed for every significant $\left|\delta^*_{h,i}\right|$ such that $|\delta^*_\infty| \geq \delta_\epsilon$. In this alternative algorithm, the complexity

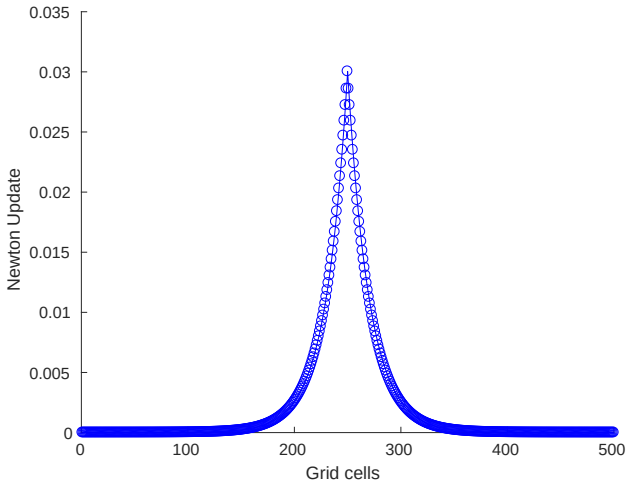


Figure 5.1: Monotonic Newton update for a single nonzero discrete residual $R_{h,i=250}$.

of flagging the nonzero support set for the analytical estimate of the Newton update is reduced to $O(M_{NNZ})$ instead of $O(N * M_{NNZ})$, where N is the number of control volumes and M_{NNZ} is the number of control volumes exhibiting significant Newton update calculated using just the diagonal blocks of the jacobian matrix.

Hence assuming that $B^{-1}_{max}A_{min}$ is diagonalizable, instead of solving Equation 5.2.11 for the coupled system, we can get decoupled solutions for each component. The decoupled equations reduce to the form given in Equation 3.5.21 and the radius can be calculated in a similar way as shown in Equation 3.5.24.

5.3 Algorithm

In the previous section the mathematical framework is developed such that the estimated nonzero support set for the infinite Newton iteration is conservative to the numerical iterate obtained from the fully implicit simulator. In this section, a step by step computational algorithm is presented with details pertaining to the computation of the coefficient matrices and evaluation of the radii.

5.3.1 *Step 1: Evaluate* **B** *and A*

The coefficient matrices are computed over the entire domain for the first iteration and thereafter the calculation is localized. **B** is a matrix with vector coefficients which is homogenized first by taking the supremum over the spatial domain for each vectorial direction then by taking the supremum over the components.

$$B_{max} = \sup\left(\sup_{x\in\Omega} B(x)\hat{i}, \sup_{x\in\Omega} B(x)\hat{j}, \sup_{x\in\Omega} B(x)\hat{k}\right),$$

where $\hat{i}$, $\hat{j}$ and $\hat{k}$ are unit directional vectors for a three dimensional system. Matrix A is composed of scalar spatially variable functions that consists of accumulation and source terms. For a conservative radius in Equation 3.5.24, infimum of the spatially variable A matrix has to be evaluated by

$$A_{min} = \inf_{x\in\Omega}(A(x)).$$

In the case where gravity is neglected, both matrices **B** and A are block diagonal without any off block diagonal entries. While comparing matrices to compute the supremum and infimum in the above cases, the real part of the eigenvalues are considered. A matrix is considered to be larger than another only if all the eigenvalues of the former are larger than the latter. Similarly, matrix $M < N$, if $(Eigs(M) < Eigs(N))$.

5.3.2 *Step 2: Determine* M_{NNZ} *set*

In equation Equation 5.2.12, $\left|\delta^*_{h,i}\right| < \delta_\epsilon$ results in $|\delta^*_\infty(r)| < \delta_\epsilon$ and hence considered zero as it is below the convergence tolerance. Instead of calculating the radius for every control volume in the domain, a set is created which contain control volumes that exhibit $\left|\delta^*_{h,i}\right| \geq \delta_\epsilon$. The radius is then calculated only for these M_{NNZ} control volumes and the union of the flagged domains form the reduced linear system. The mathematical derivation presented in the previous sections consider a decoupled elliptic-hyperbolic system while the implementation of this process is to fully implicit coupled simulators. To capture the effect of the pressure variable onto the hyperbolic state variables, to an extent, a coupled block

diagonal system is considered for the calculation of $\delta^*_{h,i}$. For each control volume, i, the dense submatrix will be of size $nc+1$ and is given by

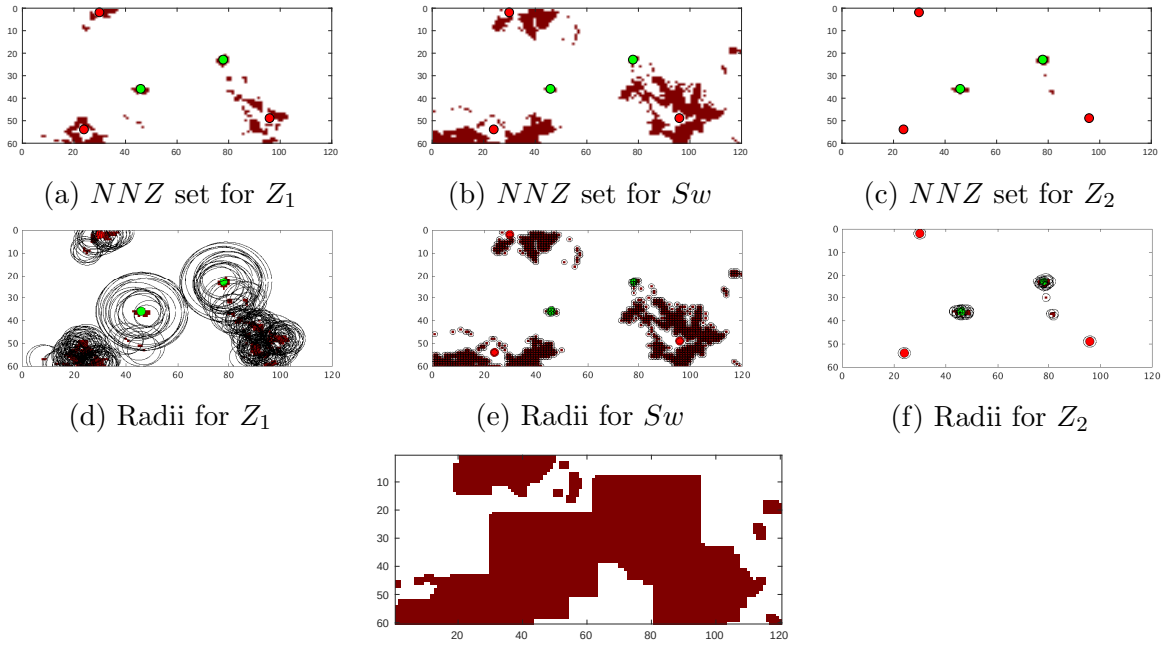
$$\begin{bmatrix} \partial R_{h,i,P}/\partial P & \partial R_{h,i,P}/\partial Sw & \dots & \partial R_{h,i,P}/\partial z_{nc-1} \\ \partial R_{h,i,Sw}/\partial P & \partial R_{h,i,Sw}/\partial Sw & \dots & \partial R_{h,i,Sw}/\partial z_{nc-1} \\ \vdots & \vdots & \vdots & \vdots \\ \partial R_{h,i,z_{nc-1}}/\partial P & \partial R_{h,i,z_{nc-1}}/\partial Sw & \dots & \partial R_{h,i,z_{nc-1}}/\partial z_{nc-1} \end{bmatrix} \begin{bmatrix} \delta^*_{h,i,P} \\ \delta^*_{h,i,Sw} \\ \vdots \\ \delta^*_{h,i,z_{nc-1}} \end{bmatrix} = - \begin{bmatrix} R_{h,i,P} \\ R_{h,i,Sw} \\ \vdots \\ R_{h,i,z_{nc-1}} \end{bmatrix}$$

The hyperbolic components, $\left|\delta^*_{h,i,Sw}\right| \geq \delta_\epsilon$ and $\left|\delta^*_{h,i,z_c}\right| \geq \delta_\epsilon$, form the support set for the radius calculation. All the pressure updates, $\delta^*_{h,i,P}$, are considered in the active set.

5.3.3 Step 3: Evaluate radius and flag

For each entry in the M_{NNZ} set, Equation 3.5.24 is used to evaluate the radius for each component. This radius is used to draw a circle around the control volume containing $\left|\delta^*_{h,i}\right| > \delta_\epsilon$ in the discrete domain. Union of all the circles form the localized linear system which is solved at the current iteration. In figure 5.2, the first step is to calculate the M_{NNZ} set as described in Step 2 of this section. Figures 5.2a, 5.2c and 5.2b show the nonzero residual set for the component C_1, C_2 and water saturation, respectively. Similar sets are obtained for each component in the hyperbolic system. In these figures, green markers are the injector wells while the red circles are the producers. Figures 5.2d, 5.2e and 5.2f show the circles obtained for each entry in the nonzero set for individual components. The magnitude of the radius will depend on the magnitude of $\delta^*_{h,i}$ and the values of the coefficients in matrices A and B. Figure 5.2g shows the final flagged region obtained after the union of all the circles computed in the previous step. The full linear system is reduced to this colored domain for every hyperbolic component to maintain the consistency between the material balance equations.

This algorithm is applied to several numerical test cases and compared with the traditional full Newton's method. The following sections explore the robustness and efficiency of the mathematical framework developed in this work.



(a) NNZ set for Z_1 (b) NNZ set for Sw (c) NNZ set for Z_2

(d) Radii for Z_1 (e) Radii for Sw (f) Radii for Z_2

(g) The colored region shows the linear system that needs to be solved for the current iteration

Figure 5.2: Evaluation of M_{NNZ} set for all the components, followed by radius computation for each nonzero control volume. The final flag is obtained by taking the union of all the circles obtained in Figures 5.2d, 5.2e and 5.2f.

5.4 Preconditioner decoupling

Equation 3.5.24 is obtained for a decoupled elliptic-hyperbolic system. In practice, this method is applied to coupled fully implicit simulations and hence results in deviation from the fully implicit convergence rate. This discrepancy is addressed by preconditioning the full jacobian system using two different preconditioners. For a fully implicit problem, these preconditioners will not affect the convergence. In the case of localized solvers, this decoupling might prove beneficial because a change in the local compositional field will produce a small change in the global pressure field. If these state variables are decoupled, the local effects on a global scale will be limited thereby improving the convergence characteristics.

5.4.1 Full decoupling

The first preconditioner is similar to the first stage of CPR (Constrained Pressure Residual) type linear solver. There are many instances of this type of preconditioner when

specialized solvers are to be employed to exploit the distinct properties of the evolution of state variables e.g. [12]. The fully coupled system, in matrix form, is

$$A\delta = -R. \tag{5.4.1}$$

Let the subscripts p and c stand for the elliptic (pressure) and hyperbolic (compositions) state variables of the governing equation. R is the residual vector and A_{xy} depicts the derivatives of the x equation with respect to the y variable. The above matrix equation can be rewritten as

$$\begin{bmatrix} A_{pp} & A_{pc} \\ A_{cp} & A_{cc} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix} = - \begin{bmatrix} R_p \\ R_c \end{bmatrix},$$

where R_c is the system of hyperbolic residual equations. There is one pressure equation and N_c composition equations. Multiplying the second row by $A_{pc}A_{cc}^{-1}$ and subtracting it from the first row we obtain

$$\begin{bmatrix} A_{pp} - A_{pc}A_{cc}^{-1}A_{cp} & 0 \\ A_{cp} & A_{cc} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix} = - \begin{bmatrix} R_p - A_{pc}A_{cc}^{-1}R_c \\ R_c \end{bmatrix}.$$

Hence the modified pressure equation becomes,

$$(A_{pp} - A_{pc}A_{cc}^{-1}A_{cp})\delta_p = -(R_p - A_{pc}A_{cc}^{-1}R_c). \tag{5.4.2}$$

By this method, the pressure equation is essentially decoupled from the compositional state. Change in the local physics will only slightly affect the global pressure state. This is required because the localization algorithm is applied only to the hyperbolic system of equations.

5.4.2 IMPES decoupling

The second preconditioner utilizes the IMPES pressure equation concept presented

in [16]. Similar to the full decoupling preconditioner, the coupled system can be written as

$$A_1\delta + A_2\delta = -R, \tag{5.4.3}$$

where A_1 and A_2 represent the accumulation and transmissibility parts of the jacobian. Using the notation developed in the previous section, the above matrix can be decomposed in the following way:

$$\begin{bmatrix} A_{pp,1} & A_{pc,1} \\ A_{cp,1} & A_{cc,1} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix} + \begin{bmatrix} A_{pp,2} & A_{pc,2} \\ A_{cp,2} & A_{cc,2} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix} = - \begin{bmatrix} R_p \\ R_c \end{bmatrix}.$$

Multiplying the second row by $A_{pc,1}A_{cc,1}^{-1}$ and subtracting it from the first row we obtain

$$\begin{bmatrix} A_{pp,1} - A_{pc,1}A_{cc,1}^{-1}A_{cp,1} & 0 \\ A_{cp,1} & A_{cc,1} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix}$$
$$+ \begin{bmatrix} A_{pp,2} - A_{pc,1}A_{cc,1}^{-1}A_{cp,2} & A_{pc,2} - A_{pc,1}A_{cc,1}^{-1}A_{cc,2} \\ A_{cp,2} & A_{cc,2} \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_c \end{bmatrix} = - \begin{bmatrix} R_p - A_{pc,1}A_{cc,1}^{-1}R_c \\ R_c \end{bmatrix}.$$

Hence the modified pressure equation becomes

$$(A_{pc,1}A_{cc,1}^{-1}A_{cp,1} + A_{pp,2} - A_{pc,1}A_{cc,1}^{-1}A_{cp,2})\delta_p + (A_{pc,2} - A_{pc,1}A_{cc,1}^{-1}A_{cc,2})\delta_c = -(R_p - A_{pc,1}A_{cc,1}^{-1}R_c). \tag{5.4.4}$$

By using this preconditioner the pressure equation is decoupled from the compositional state variables in the accumulation terms. The coupling is still allowed through the modified transmissibilities of the original conservation equation. Only local fluxes will affect the global pressure state, while the effect of compressibility is eliminated.

5.5 Results

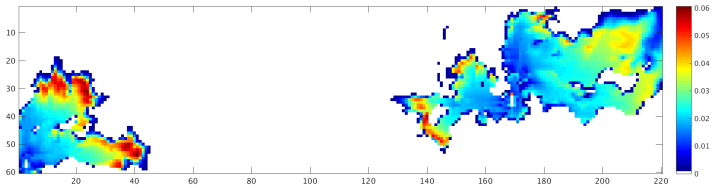
5.5.1 Numerical locality

The first and major observation was the extent of locality, if any, present in the full numerical model. As an example, the forty-eight layer of the SPE10 comparative study case is chosen with two gas injector wells and three producers. Figure 5.3 shows five consecutive Newton iteration snapshots at time 50 days of simulation. In each figure, the solution of the reduced linear system is shown. The first iteration, Figure 5.3a, shows the maximum change with the largest magnitude of the Newton update near the fronts. As the iterations proceed, Figures 5.3b to 5.3e, the locality in the Newton iterates increase and the magnitude of the change decreases. Figure 5.3f is obtained by the summation of the five iterations and shows the change in the state variable over one time step, i.e. n to $(n+1)$. The locality within a Newton process for the solution of a time-step is equivalent to the sparsity of the Newton updates over the corresponding number of iterations. The level of locality that is present is related to, but quite different from the locality that will be present over the entire time-step. While the time-step update may be dense, the individual Newton updates may show a relatively large degree of sparsity.

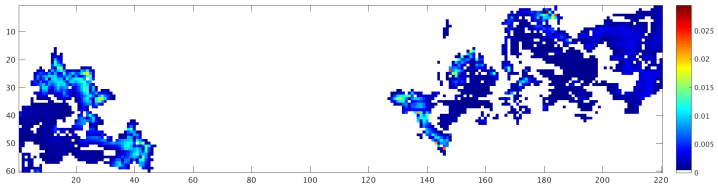
5.5.2 Preconditioner results

Figure 5.4 and 5.5 show the comparison between different solution strategies in terms of the iterations required for two different cases of SPE10 slices. The simulation is run for several timesteps using the full Newton solver without preconditioner, localized solver without preconditioner, and localized solver with full and IMPES preconditioners. Figure 5.4 describes the case of channelized permeability field while Figure 5.5 is the simulation on a Gaussian permeability and porosity field. The circle markers (black line) show the iterations required for the full solution of the jacobian system while the diamond markers (green line) are the result of local computation of Newton updates without any preconditioner. This discrepancy in the iteration count is the result of application of the estimate obtained from decoupled flow and transport equations onto a fully coupled problem.

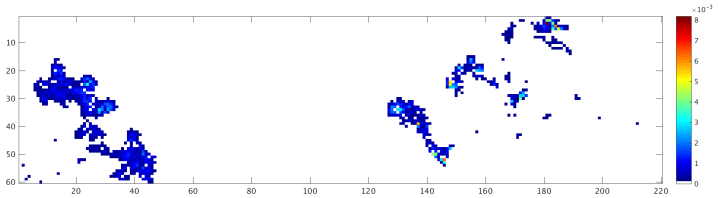
Modification of the pressure equation as shown in Equation 5.4.2 and 5.4.4 results in



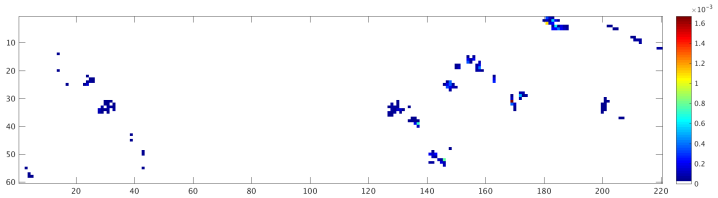
(a) Iteration 1



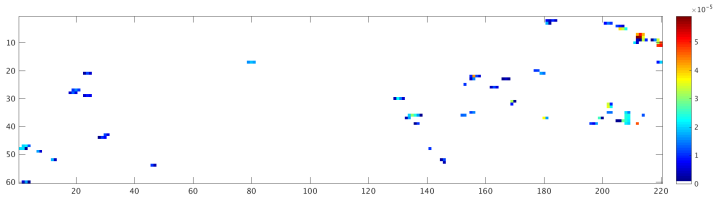
(b) Iteration 2



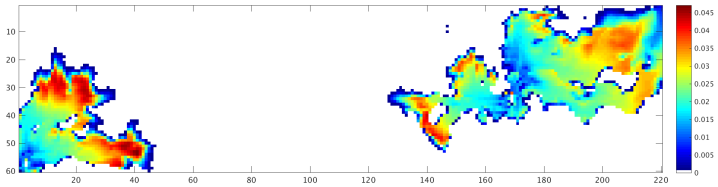
(c) Iteration 3



(d) Iteration 4



(e) Iteration 5



(f) $\left|u^{(n+1)} - u^{(n)}\right|$

Figure 5.3: This figure shows five consecutive Newton iterations and the final state at time level $(n+1)$. In Figures 5.3a to 5.3e, the colored regions show $|\delta_{h,Z_1}| \geq \delta_{TOL}$.

the plus markers (blue line) and the square markers (red line), respectively. In the first case, the iteration count of the full preconditiner application is exactly the same as that of full solution strategy. IMPES preconditiner improves the result and has marginal discrepancy in the iteration counts from that of the fully implicit scheme. In the second case, the preconditioners take just a couple iterations more than solving the full jacobian.

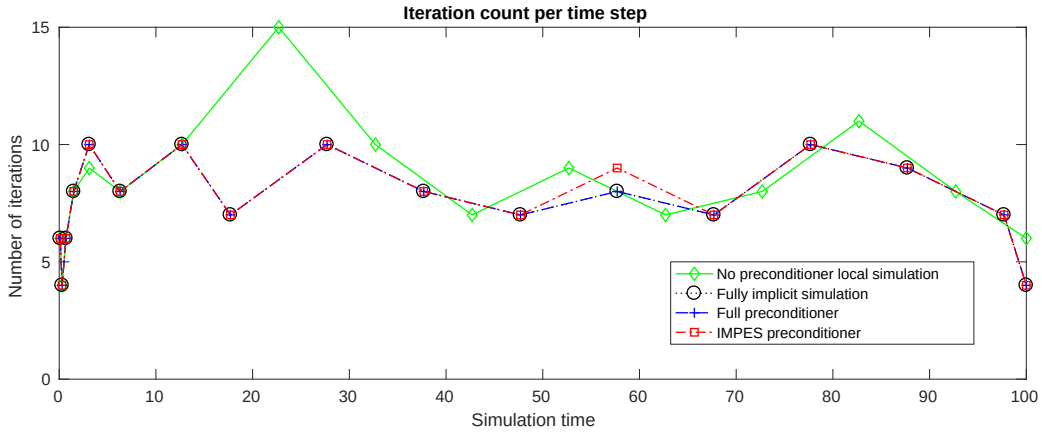


Figure 5.4: Comparisons of number of iterations for different strategies applied to SPE10 84^{th} layer.

The mathematical framework based on decoupled flow and transport equations can be applied to fully coupled compositional models if similar decoupling preconditioners are applied.

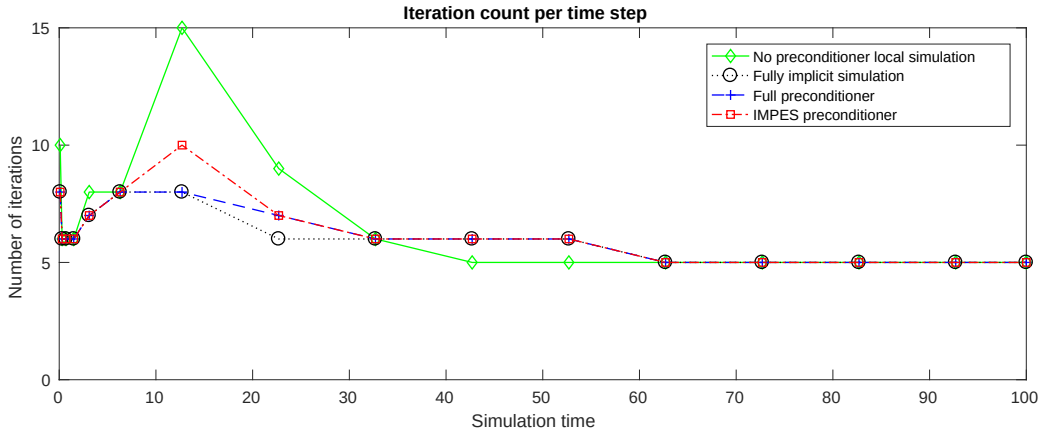


Figure 5.5: Comparisons of number of iterations for different strategies applied to SPE10 1^{st} layer.

5.5.3 Computational examples

In this section the localization algorithm is implemented in a two-phase EOS compositional simulator for the different injection and depletion cases. All the comparisons are carried out with fully implicit simulator which solves the full jacobian system to obtain the numerical solution. The time stepping strategy is similar in all the cases with a maximum time-step size of 10 days. Along with the percentage of domain flagged for each test case, the speedup is presented in terms of complexity. The state of the art linear solver has a complexity of $NlogN$, where $N \times N$ is the size of the matrix. Hence the speedup, S, is given by

$$S = \frac{C \times n_{iter} N log(N)}{C \times n_{iter}^{loc} pN log(pN)}, \tag{5.5.1}$$

where n_{iter} and n_{iter}^{loc} are the total number of iterations required for the full Newton method and the localized algorithm. p is the average percentage of the domain flagged over the entire simulation and C is a constant which will be required when accounting for the cost of flagging.

5.5.4 Case 1: Injection and depletion

To test the complexity introduced by gas injection, component C_1 is added at constant injection rate. In this case the surface injection rate is 1000 $Mscf/day$ and the bottomhole well pressure is 800 psi which is below the bubble point pressure for this thermodynamic system. The initial reservoir pressure is 4000 psi which is above the bubble point pressure. This case is an example of extreme drainage as within 50 days of simulation, 98% of the reservoir exhibits a two phase state. Figure 5.6 shows the percentage of domain flagged over several timesteps. An average of 50% of the reservoir was solved each iteration. The final oil and gas saturation distribution is shown in Figure 5.7.

Calculating the speedup in terms of the complexity of the linear solver, a 2.5 folds faster solution is obtained. The speedup is proportional to the locality present within the computational domain along with the initial number of nonzero residual entries.

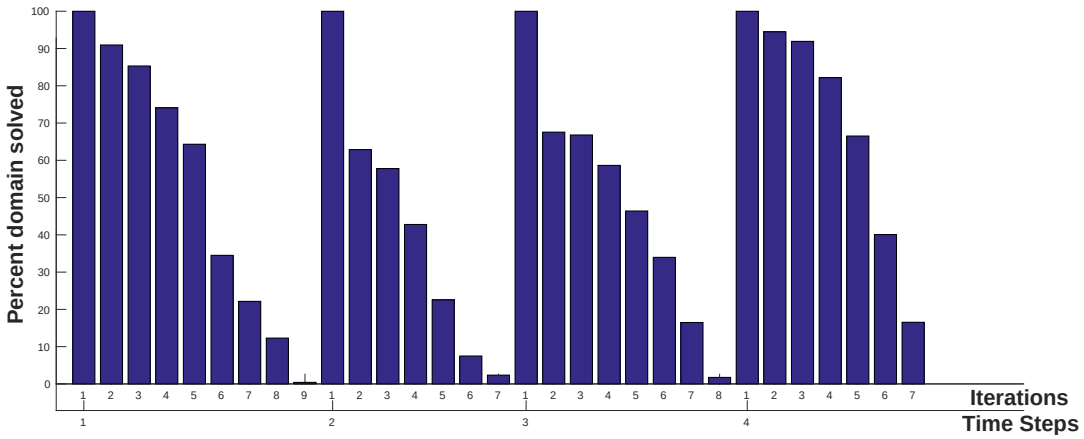


Figure 5.6: Percent domain flagged for case 1. Average percent solved = 50%.

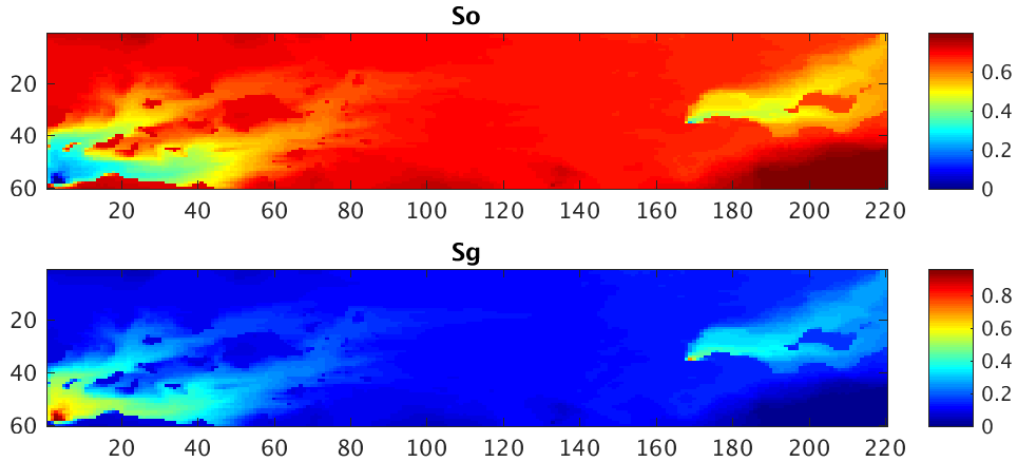


Figure 5.7: Final oil and gas saturation distribution for case 1.

5.5.5 Case 2: Injection and depletion

In practice, case 1 will seldom be encountered. The percentage of reservoir resulting in two phase state will be considerably smaller than in the previous case. To create a more local case, the bottomhole flowing pressure is set at 3000 *psi* which makes the drawdown pressure to be 1000 *psi* and the gas injection rates are set at $500 Mscf/day$. In this case, most of the reservoir stay above bubble point. After a simulation of 100 days, the percent domain in the two phase state is 14%. In this case the average percent domain that is flagged during the linear solution step is around 14%, as shown in Figure 5.8. Due to the

local update in the transport variables, the speedup obtained in this case is close to 13 folds.

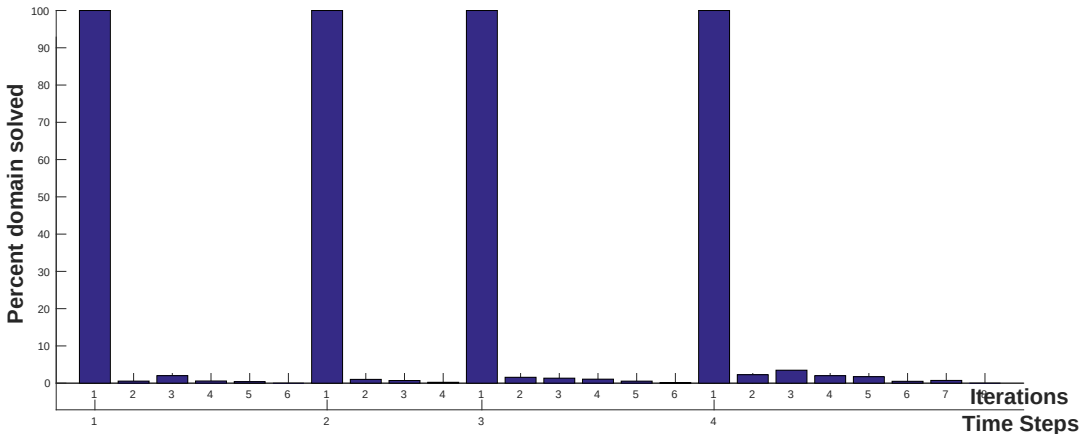


Figure 5.8: Percent domain flagged for case 2. Average percent solved = 14%.

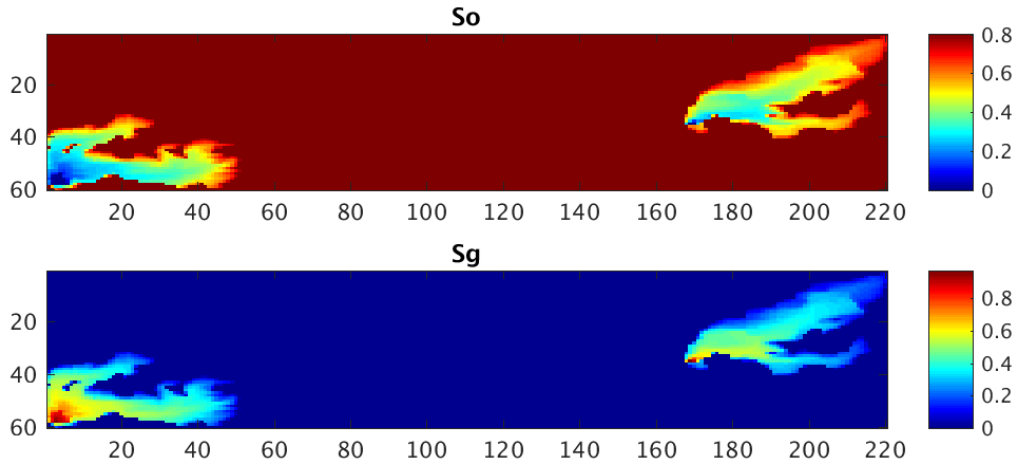


Figure 5.9: Final oil and gas saturation distribution for case 2.

Figure 5.9 shows the final saturation distribution after a simulation of 100 days. It can be observed that the proposed algorithm adapts to the underlying heterogeneity efficiently.

5.5.6 Case 3: Depletion with gravity

In the previous cases, gravity was neglected. This test case is setup to study the depletion scenario which also addressed the issue of phase appearance. The reservoir is not initialized to the gravity-capillarity equilibrium. Even without initializing the reservoir,

there is considerable locality within the Newton process. Bottomhole pressures are set at 800 *psi* and the initial pressure was maintained at 3000 *psi*. Gravity is acting in the positive x-direction and Figure 5.10 shows the final distribution of phase saturations.

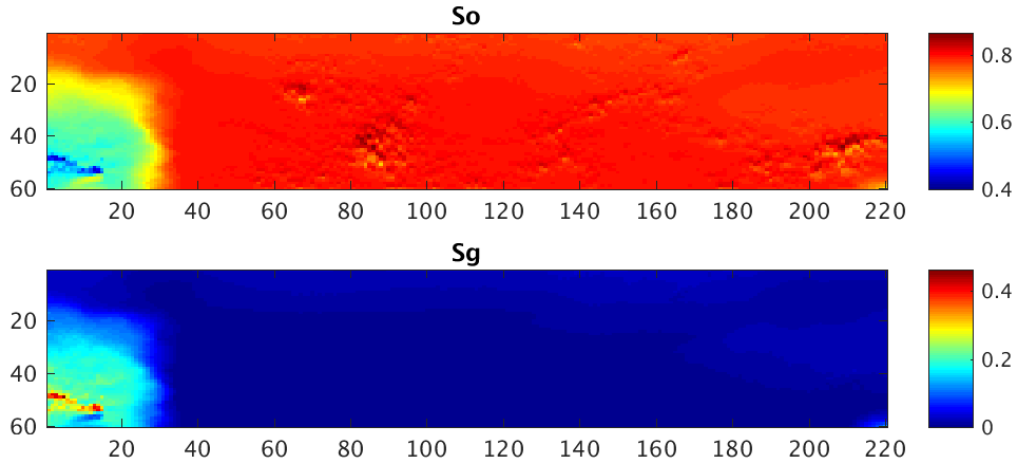


Figure 5.10: Final oil and gas saturation distribution for case 3.

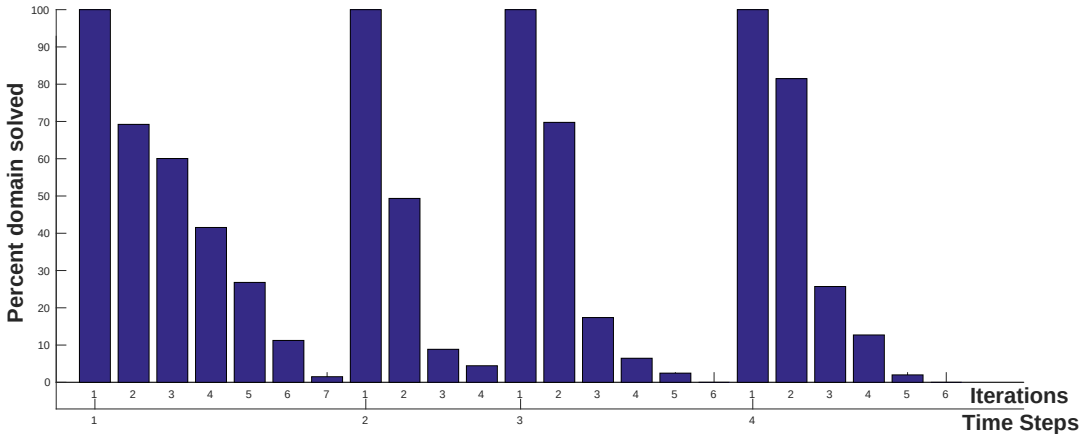


Figure 5.11: Percent domain flagged for Case 3. Average percent solved = 24%.

The average percentage of domain flagged over the course of the simulation is around 24%, as shown in Figure 5.11. The percent domain exhibiting a two phase state is close to 57%. In terms of the linear computational effort, the speedup obtained is 6.4 folds.

5.5.7 Case 4: Injection and depletion with gravity

In this final case, injection and depletion are tested with gravity acting in the positive x-direction. There is one injection well with gas injection rate of 1000 $Mscf/day$ and two production wells with the bottomhole pressure set at 1000 psi. The drawdown pressure is 3000 psi. At the end of 50 days, the percentage of domain in the two phase state is around 15%. The speedup obtained here is 4.5 times with an average of 27% domain being flagged by the localization algorithm (Figure 5.12).

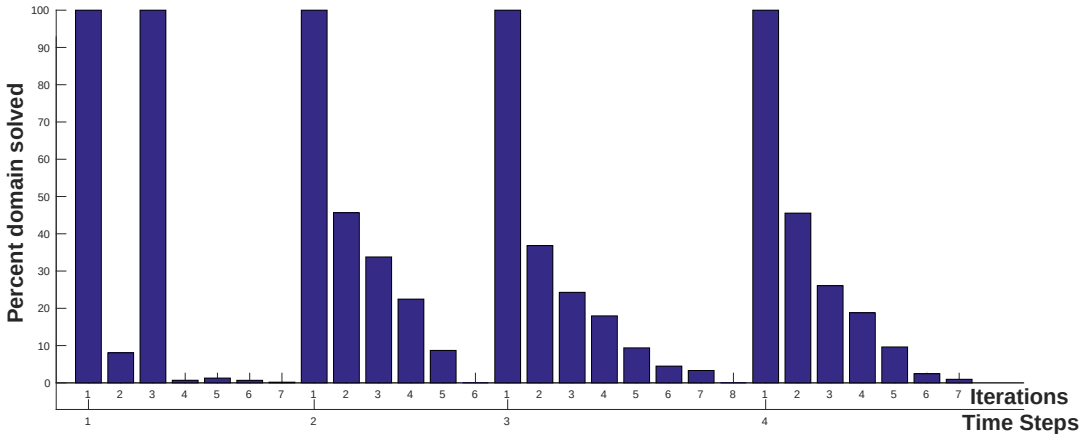


Figure 5.12: Percent domain flagged for Case 4. Average percent solved = 27%.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

An algorithmic approach and associated theory are developed to enable the localized solution of linear systems that arise over the course of the nonlinear solution process for a sequential implicit time-step. The methods are conservative and reproduce the Newton updates exactly, thereby preserving the nonlinear convergence rate. The degree of locality for flow varies across iterations, and is a strong function of variable screening parameter which depends directly on the compressibility of the system and the numerical time step size and indirectly on the underlying heterogeneity.

This mathematical framework is used to predict the evolution of Newton updates during the simulation of fluid flow in porous media. Two approximations are committed to facilitate the computation of these estimates. The first is a homogenization process and the second concerns the geometry of the simulation domain. The theory that justifies these assumptions to lead to conservative estimates in the sense of support is presented in Chapter 2. The numerical examples presented universally show strong empirical evidence of this important property. Using the proposed localized solver, we can obtain simulations that are many folds faster depending on the underlying heterogeneity and complexity. We also show that heuristic methods may fail to capture the support set conservatively and thus affect the solution process adversely. Further research includes exploring better and advanced homogenization strategies for pressure equation and adapting the localized solver for multidimensional coupled flow and transport problems.

In the case of sequential-implicit simulation of compositional models, the infinite-dimensional Newton problem reduces to a scalar elliptic BVP of the form in Equation 3.4.1 for flow, algebraic equations for thermodynamic constraints, and a hyperbolic system of the

form

$$\nabla\delta + A\delta = b,$$

for transport variables, similar to the one shown in Equation 5.2.10. In the above equation, A is the coefficient matrix and b is the residual vector. The solution of the above equation is given by the matrix exponential form, where the matrix exponent can be evaluated using a multitude of methods such as Putzer algorithm [45, 56].

A semi-heuristic algorithm is presented in this work that extends the application of the localized solver to a system of hyperbolic equations. Chapter 5 presents the algorithm and some numerical examples for a fully-implicit compositional model. This heuristic extension includes the application of two decoupling methods that improves the convergence rate of the proposed localized Newton method. While the results for this extension are encouraging, further research is required to develop strong theoretical justifications for the same. The numerical examples in this work are targeted at testing the proposed methodology, but they also indicate a promising opportunity in reservoir simulation. Adaptive solvers such as the proposed method can be applied in many other contexts including multiscale simulation.

The extension of this work to general fully-implicit simulation of coupled flow and transport is a topic of interest. The theory connecting the finite and projected infinite problems carries through to the case of systems of equations. The principle challenge is then to derive analytical solutions to the systems of BVP. Solutions to constant coefficient elliptic BVP are the subject of numerous articles. In principle, one amenable approach is the use of a Schur complement in the Banach space.

Another application of the theory presented is in devising new preconditioning strategies akin to the CPR method but relying purely on highly scalable boundary integral preconditioners for the infinite problem.

Nomenclature

$|\cdot|$ Absolute norm.

$\bar{\gamma}_p$ Mass density of phase p (lb/ft^3).

Δr Gridcell size in radial coordinates.

Δt Time-step size.

δ^* Estimate to the infinite dimensional Newton update.

δ_ϵ Cutoff value of Newton update. In this work its taken to be 10E-06.

δ_∞^ν Continuous Newton update for iteration ν.

δ_h Newton update in the discrete space (finite dimensional Newton update).

δ_p Newton update for the pressure variable.

δ_s Newton update for the saturation variable.

ϵ_m Machine precision number.

$\frac{\partial u}{\partial \nu}$ Gradient along the outward-oriented unit-normal on $\partial\Omega$.

$\mathbf{f} : \Omega \times D \to \mathbb{R}^d$ inviscid flux.

$\mathbf{G} : \mathbb{R}^d \to \mathbb{R}^d$ viscous flux coefficient operator.

$\mathbf{I}_0$ Zero order modified Bessel's function of the first kind.

$\mathbf{I}_1$ First order modified Bessel's function of the first kind.

$\mathbf{K}$ Absolute permeability tensor.

$\mathbf{L}_0$ Zero order Struve function.

$\mathbf{L}_1$ First order Struve function.

μ_p Viscosity of phase p (cP).

∇D Difference in the elevation from a datum (ft).

$\|\cdot\|_0$ l_0 norm or the cardinality of a function.

Ω Bounded spatial domain with $\Omega \subset \mathbb{R}^d$ and $d \in \{1, 2, 3\}$.

$\partial\Omega$ Boundary of the solution domain.

ϕ Porosity of the rock.

ρ_p Molar density of phase p $(lbmol/ft^3)$.

$\tilde{h}$ Corrector function as a function of r evaluated at r^*.

$a : \Omega \times D \to \mathbb{R}$ change in accumulation over the time-step.

B_∞ Semi-discrete boundary condition.

e Euler's number.

$G(r, r^*)$ Green's function as a function of r and r^*.

H_{v_i} Heaviside distribution around v_i.

k_{rp} Relative permeability of phase p.

n_c Number of hydrocarbon components.

n_p Number of phases.

N_{V_i} Number of control volumes.

$q_{c,p}$ Sink or source term for component c in phase p.

r Radial position vector ($\mathbb{R}$).

R_∞ Semi-discrete residual function.

S_p Saturation of phase p.

V_b Bulk volume (ft^3).

V_i Control volume.

$w : \Omega \times D \to \mathbb{R}$ net sink term.

x Position vector ($\mathbb{R}^N$). For three dimensional problems, N = 3.

$x_{c,p}$ Mole fraction of component c in phase p.

BIBLIOGRAPHY

[1] E.L. Allgower, K. Böhmer, F.A. Potra, and W.C. Rheinboldt. A mesh-independence principle for operator equations and their discretizations. *Society of Industrial Applied Mathematics Journal on Numerical Analysis*, 23(1):160–169, 1986.

[2] E.L. Allgower, K. Böhmer, F.A. Potra, and W.C. Rheinboldt. A mesh-independence principle for operator equations and their discretizations. *Society of Industrial Applied Mathematics Journal on Numerical Analysis*, 23(1):160–169, 1986.

[3] I.K. Argyros. A mesh-independence principle for operators equations and the steffensen method. *Korean Journal of Computational & Applied Mathematics*, 4(2):263–280, 1997.

[4] F. Armero and J.C. Simo. A new unconditionally stable fractional step method for non-linear coupled thermomechanical problems. *International Journal for numerical methods in Engineering*, 35(4):737–766, 1992.

[5] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2):151–167, 1997.

[6] U.M. Ascher, S.J. Ruuth, and B.T. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

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

[8] A. Behie. Comparison of nested factorization, constrained pressure residual, and incomplete factorization preconditionings. In *SPE-13531-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 1985.

[9] H. Berestycki, L. Nirenberg, and S.R.S Varadhan. The principal eigenvalue and maximum principle for second order elliptic operators in general domains. *Communications on Pure and Applied Mathematics*, 47:47 – 92, 1994.

[10] X.C. Cai, D.E. Keyes, and L. Marcinkowski. Non-linear additive schwarz preconditioners and application in computational fluid dynamics. *International journal for numerical methods in fluids*, 40(12):1463–1470, 2002.

[11] J. Campos, J. Mawhin, and R. Ortega. Maximum principles around an eigenvalue with constant eigenfunctions. *Communications in Contemporary Mathematics*, 10:1243 – 1257, 2008.

[12] H. Cao, H.A. Tchelepi, J. Wallis, and H. Yardumian. Parallel scalable unstructured cpr-type linear solver for reservoir simulation. In *SPE Annual Technical Conference and Exhibition*, 2005.

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

[14] M.A. Christie and M.J. Blunt. Tenth spe comparative solution project: A comparison of upscaling techniques. *Society of Petroleum Engineers Reservoir Evaluation & Engineering*, 4(4):308–317, 2001.

[15] K.H. Coats. A note on impes and some impes-based simulation models. *SPE Journal*, 5(03):245–251, 2000.

[16] K.H. Coats. A note on IMPES and some IMPES-based simulation models. *Society of Petroleum Engineers*, 5:245–251, 2000.

[17] Romain De Loubens, Amir Riaz, and Hamdi A Tchelepi. Error analysis of an adaptive implicit scheme for hyperbolic conservation laws. *SIAM Journal on Scientific Computing*, 31(4):2890–2914, 2009.

[18] A.H. Dogru, L.S.K. Fung, U. Middya, T. Al-Shaalan, and J.A. Pita. A next-generation parallel reservoir simulator for giant reservoirs. In *SPE-119272-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 2009.

[19] M.S. Espedal and K.H. Karlsen. *Numerical solution of reservoir flow models based on large time step operator splitting algorithms*. Springer, 2000.

[20] Lawrence C. Evans. *Partial differential equations*. Graduate studies in mathematics. American Mathematical Society, Providence (R.I.), 1998.

[21] Schlumberger Geoquest. Eclipse 100 reference manual. *Schlumberger Geoquest*, 2001.

[22] M. Gerritsen and J.V. Lambers. Integration of local–global upscaling and grid adaptivity for simulation of subsurface flow in heterogeneous formations. *Computational Geosciences*, 12(2):193–208, 2008.

[23] S. Gries, K. Stüben, G. L. Brown, D. Chen, and D.A. Collins. Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations. *SPE Journal*, 2907, 2004.

[24] B. Gustafsson, H. Kreiss, and J. Oliger. *Time dependent problems and difference methods*, volume 24. John Wiley & Sons, 1995.

[25] R.D. Hornung and J.A. Trangenstein. Adaptive mesh refinement and multilevel iteration for flow in porous media. *Journal of computational Physics*, 136(2):522–545, 1997.

[26] Ulrich Hornung. *Homogenization and porous media*, volume 6. Springer Science & Business Media, 2012.

[27] F.N. Hwang and X.C. Cai. A parallel nonlinear additive schwarz preconditioned inexact newton algorithm for incompressible navier–stokes equations. *Journal of Computational Physics*, 204(2):666–691, 2005.

[28] R.I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1):40–65, 1986.

[29] Patrick Jenny, Hamdi A. Tchelepi, and Seong H. Lee. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with s-shaped flux functions. *J. Comput. Phys.*, 228(20):7497–7512, November 2009.

[30] K.H. Karlsen, K.A. Lie, J.R. Natvig, H.F. Nordhaug, and H.K. Dahle. Operator splitting methods for systems of convection–diffusion equations: nonlinear error mechanisms and correction strategies. *Journal of Computational Physics*, 173(2):636–663, 2001.

[31] C.T. Kelley and E.W. Sachs. Mesh independence of newton-like methods for infinite dimensional problems. *Journal of Integral Equations and Applications*, 3(4):549–573, 1991.

[32] Felix Kwok and Hamdi Tchelepi. Potential-based reduced newton algorithm for nonlinear multiphase flow in porous media. *J. Comput. Phys.*, 227(1):706–727, November 2007.

[33] S. Lacroix, Yu. Vassilevski, J. Wheeler, and M. Wheeler. Iterative solution methods for modeling multiphase flow in porous media fully implicitly. *SIAM J. Sci. Comput.*, 25(3):905–926, March 2003.

[34] Boxiao Li and Hamdi A. Tchelepi. Unconditionally convergent nonlinear solver for multiphase flow in porous media under viscous force, buoyancy, and capillarity. *Energy Procedia*, 59:404 – 411, 2014.

[35] Knut-Andreas Lie, Halvor Møll Nilsen, Atgeirr Flø Rasmussen, Xavier Raynaud, et al. Fast simulation of polymer injection in heavy-oil reservoirs on the basis of topological sorting and sequential splitting. *SPE Journal*, 19(06):991–1, 2014.

[36] P. Lu and B.L. Beckner. An adaptive newtons method for reservoir simulation. In *SPE-141935-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 2011.

[37] A. Manea, J. Sewall, and H.A. Tchelepi. Parallel multiscale linear solver for highly detailed reservoir models. *SPE Journal*, 2016.

[38] Roland Masson, Philippe Quandalle, Stephane Requena, and Robert Scheichl. Parallel preconditioning for sedimentary basin simulations. *Lecture Notes in Computer Science*, 2907:93–102, 2004. Series volume title: Large-Scale Scientific Computing: 4th International Conference, LSSC 2003, Sozopol, Bulgaria, June 4-8, 2003. Revised Papers. Part II Print ISBN: 978-3-540-21090-0 Online ISBN: 978-3-540-24588-9.

[39] J.A. Meyerink. Iterative methods for the solution of linear equations based on incomplete block factorization of the matrix. In *SPE-12262-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 1983.

[40] O. Møyner. Nonlinear solver for three-phase transport problems based on approximate trust regions. In *14th European conference on the mathematics of oil recovery*, 2014.

[41] Jostein R Natvig and Knut-Andreas Lie. Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements. *Journal of Computational physics*, 227(24):10108–10124, 2008.

[42] J.H. Peery and E.H. Herron. Three-phase reservoir simulation. *Journal of Petroleum Technology*, 21:211–220, 1969.

[43] M. H. Protter and H. Weinberger. *Maximum Principles in Differential Equations*. Prentice Hall, Englewoods Cliffs, 1967.

[44] P. Pucci and J. Serrin. *The Maximum Principle*. Birkhäuser, 2007.

[45] E.J Putzer. Avoiding the jordan canonical form in the discussion of linear systems with constant coefficients. *American Mathematical Monthly*, 73(1):2 – 7, 1966.

[46] R.D. Richtmyer and K.W. Morton. Difference methods for initial-value problems. *Malabar, Fla.: Krieger Publishing Co.,— c1994, 2nd ed.*, 1, 1994.

[47] TF Russell et al. Stability analysis and switching criteria for adaptive implicit methods based on the cfl condition. In *SPE Symposium on Reservoir Simulation*. Society of Petroleum Engineers, 1989.

[48] S.M. Sheth and R.M. Younis. Asynchronous multirate newton - a class of nonlinear solver that adaptively localizes computation. In *14th European conference on the mathematics of oil recovery*, 2014.

[49] S.M. Sheth and R.M. Younis. Localized solvers for general full-resolution implicit reservoir simulation. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[50] J.O. Skogestad, E. Keilegavlen, and J.M. Nordbotten. Domain decomposition strategies for nonlinear flow problems in porous media. *Journal of Computational Physics*, 2012.

[51] A.G Spillette, J.G. Hillestad, and H.L. Stone. A high-stability sequential solution approach to reservoir simulation. *SPE*, 4542, 1973.

[52] B. Sportisse. An analysis of operator splitting techniques in the stiff case. *Journal of Computational Physics*, 161(1):140–168, 2000.

[53] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(12):281 – 309, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[54] S. Sun, D. Keyes, and L. Liu. Fully implicit two-phase reservoir simulation with the additive schwarz preconditioned inexact newton method. In *Society of Petroleum Engineers Reservoir Characterisation and Simulation Conference and Exhibition*, 2013.

[55] J.A. Trangenstein. Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Advances in Water Resources*, 25(8):1175–1213, 2002.

[56] Luis Verde-Star. On linear matrix differential equations. *Advances in Applied Mathematics*, 39(3):329 – 344, 2007.

[57] J.G. Verwer, J.G. Blom, and W. Hundsdorfer. An implicit-explicit approach for atmospheric transport-chemistry problems. *Applied Numerical Mathematics*, 20(1):191–209, 1996.

[58] J. Wallis, R. Kendall, T. Little, and et. al. Constrained residual acceleration of conjugate residual methods. In *SPE-13536-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 1985.

[59] J.R. Wallis. A new high-performance linear solution method for large-scale reservoir simulation. In *SPE-25280-MS in Proceedings of the SPE Reservoir Simulation Symposium*, 1993.

[60] Xiaochen Wang and Hamdi A. Tchelepi. Trust-region based solver for nonlinear transport in heterogeneous porous media. *J. Comput. Phys.*, 253:114–137, November 2013.

[61] J.W Watts. A compositional formulation of the pressure and saturation equations. *SPE Reservoir Engineering*, 1(3):243–252, 1986.

[62] Martin Weiser, Anton Schiela, and Peter Deuflhard. Asymptotic mesh independence of newton's method revisited. *SIAM journal on numerical analysis*, 42(5):1830–1845, 2005.

[63] M.F. Wheeler and C.N. Dawson. An operator-splitting method for advection-diffusion-reaction problems. *The Mathematics of Finite Elements and Applications VI*, pages 463–382, 1987.

[64] R.M. Younis. A sharp analytical bound on the spatiotemporal locality in general two-phase flow and transport phenomena. *Procedia Computer Science*, 18:473–480, 2013.

[65] R.M. Younis, H. Tchelepi, and K. Aziz. Adaptively localized continuation-newton method–nonlinear solvers that converge all the time. *Society of Petroleum Engineers Journal*, 15(2):526–544, 2010.

[66] H. Zhou and H.A. Tchelepi. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. *SPE Journal*, 17, 2012.

APPENDIX A

TWO PHASE GOVERNING EQUATIONS AND CANONICAL FORM

The sequential implicit solution conveniently separates the solution of the pressure equation, which often elliptic in nature, from the saturation equation, which exhibits hyperbolic character. For further reading the please refer to the works of [51] and [61]. We consider the case of two-phase compressible flow in heterogeneous porous medium in multiple dimensions. Nonlinear relative permeability models and injection/production wells are considered to increase the complexity and nonlinearity of the problem. The state variables are pressure, $p(x, y, t)$ and saturation of water, $S(x, y, t)$, which are aligned with the mass conservation governing equations given by

$$\frac{\partial \phi(\rho_o S_o + \rho_w S_w)}{\partial t} = -\nabla \cdot [\rho_o \mathbf{v_o} + \rho_w \mathbf{v_w}] + q_t \tag{A.0.1}$$

and

$$\frac{\partial \phi(\rho_w S_w)}{\partial t} = -\nabla \cdot [\rho_w \mathbf{v_w}] + q_w, \tag{A.0.2}$$

where ϕ is porosity of the rock, S is the saturation, ρ_o and ρ_w are the densities of oil and water, respectively. q_t is the summation of the source terms for water and oil and $\mathbf{v}$ is the phase velocity given by the Darcy's law

$$\mathbf{v_m} = -\frac{\mathbf{K} k_{rm}}{\mu}(\nabla p_m - \rho_m g \nabla D), \tag{A.0.3}$$

where $m = o, w$, $\mathbf{K}$ is the permeability tensor, k_{rm} is the relative permeability of the phase, p_m is the phase pressure, D is the distance between two evaluation points and g is the gravity

vector. The canonical form of pressure and saturation equation can be written as

$$R_{\infty}\left(u\left(x\right)\right)=a\left(x,u\left(x\right)\right)+\Delta t\nabla\cdot\mathbf{f}\left(x,u\left(x\right)\right)+\Delta t\nabla\cdot\left[\mathbf{G}\left(x,u\left(x\right)\right)\nabla u\left(x\right)\right]+\Delta tw\left(x,u\left(x\right)\right)=0. \tag{A.0.4}$$

Subsequently, the coefficients in the above equation for

Flow become:

$$\begin{aligned}
u(x) &= p(x) \\
a(x,u(x)) &= \phi(p(x))(\rho_o(p(x))(1-S_w(x))+\rho_w(p(x))S_w(x)) \\
\mathbf{f}(x,u(x)) &= \Bigg(\left(\rho_w^2(p(x))-\rho_w(p(x))\nabla P_c(S_w(x))\right)\frac{\mathbf{K}(x)k_{rw}(S_w(x))}{\mu_w(p(x))} \\
&\qquad + \rho_o^2(p(x))\frac{\mathbf{K}(x)k_{ro}(S_w(x))}{\mu_o(p(x))}\Bigg) g\nabla D \\
\mathbf{G}(x,u(x)) &= \rho_w(p(x))\frac{\mathbf{K}(x)k_{rw}(S_w(x))}{\mu_w(p(x))}+\rho_o(p(x))\frac{\mathbf{K}(x)k_{ro}(S_w(x))}{\mu_o(p(x))} \\
w(x,u(x)) &= q_t(x,p(x),S_w(x)),
\end{aligned}$$

and for Transport become:

$$\begin{aligned}
u(x) &= S_w(x) \\
a(x,u(x)) &= \phi(p(x))\rho_w(p(x))S_w(x) \\
\mathbf{f}(x,u(x)) &= \rho_w(p(x))\frac{\mathbf{K}(x)k_{rw}(S_w(x))}{\mu_w(p(x))}(\nabla\left(p(x)-P_c(S_w(x))\right)-\rho_w(p(x))g\nabla D) \\
\mathbf{G}(x,u(x)) &= -\rho_w(p(x))\frac{\mathbf{K}(x)k_{rw}(S_w(x))}{\mu_w(p(x))}P_c'(S_w(x)) \\
w(x,u(x)) &= q_w(x,p(x),S_w(x)).
\end{aligned}$$

Relative permeability curves used in this work are quadratic for both oil and water. Density at any given pressure is calculated using the reference density and the formation volume factor. Numerical values of the reference properties are given in Table A.1. Sequential implicit simulation treats either pressure or saturation field as frozen while solving the other

Property	Value
Reference density of water	63.02 lb/ft^3
Reference density of oil	45.0 lb/ft^3
Reference viscosity of water	0.52341 cp
Reference viscosity of oil	1.177 cp
Compressibility of water	$3.0E-06$ psi^{-1}
Compressibility of oil	$1.03E-4$ (high) and $1.03E-6$(low) psi^{-1}
Compressibility of rock	$3.4E-06$ psi^{-1}

Table A.1: Fluid properties

equation. So, $a_1(p) \equiv a_1(p, S)$.

Equation A.0.4 is solved in a sequential manner, once for pressure and then for saturation. First the pressure equation is solved using frozen saturation state until convergence. The second step incorporates solution of the saturation equation assuming a frozen pressure field. Upon convergence the pressure field is updated using the new saturation state and this process continues till the desired final simulation time. The pseudo code for this sequential scheme is given in the following algorithm.

Algorithm 4: Two-phase sequential implicit scheme

Data: p^n , S^n

$p^\beta = p^n$, $S^\beta = S^n$;

do

  Solve $R_p(p^{\nu+1}; S^\nu) = 0$;

  Solve $R_s(S^{\nu+1}; p^{\nu+1}) = 0$;

while $||(p^\beta - p^{\beta-1})|| \geq \epsilon_p$ *or* $||(S^\beta - S^{\beta-1})|| \geq \epsilon_s$;

APPENDIX B

THREE PHASE BLACK OIL MODEL

The sequential implicit solution conveniently separates the solution of the pressure equation, which is often elliptic in nature, from the saturation equation, which exhibits hyperbolic character. For further reading on three phase models and sequential schemes please refer to the works of [42], [51] and [61]. We consider the case of three-phase compressible flow in heterogeneous porous medium in multiple dimensions. Nonlinear relative permeability models and injection/production wells are considered to increase the complexity and nonlinearity of the problem. The state variables are pressure, $p(x, y, t)$, saturation of water, $S_w(x, y, t)$, and saturation of gas, $S_g(x, y, t)$, which are aligned with the mass conservation equations. Due to the sequential nature of the formulation, the equation that is aligned with $p(x, y, t)$ is taken as the total mass conservation equation. The in-situ mass conservation equations are described by

$$\frac{\partial}{\partial t}(\phi(\rho_w S_w + \rho_o S_o + \rho_{go} S_o + \rho_g S_g)) + \nabla \cdot (\rho_w \mathbf{u_w} + \rho_o \mathbf{u_o} + \rho_{go} \mathbf{u_o} + \rho_g u_g) = q_t, \tag{B.0.5}$$

$$\frac{\partial}{\partial t}(\phi \rho_w S_w) + \nabla \cdot (\rho_w \mathbf{u_w}) = q_w, \tag{B.0.6}$$

$$\frac{\partial}{\partial t}(\phi(\rho_{go} S_o + \rho_g S_g)) + \nabla \cdot (\rho_{go} \mathbf{u_o} + \rho_g \mathbf{u_g}) = q_g + R_s q_o, \tag{B.0.7}$$

where S is the saturation, ρ_m is the density of oil, water or gas and R_s is the solution gas-oil ratio. q_t is the total rate of all the phases while q_o and q_g are individual oil and gas well rates. Porosity of the rock, ϕ, is a function of the reference porosity, ϕ^0, calculated at the reference pressure, p^0, and rock compressibility, c_r, which follows the geomechanical relationship given

by

$$\phi = \phi^0[1 + c_r(p - p^0)].$$

In-situ mass conservation equations can be converted to the stock tank conditions by rewriting equations B.0.5, B.0.6 and B.0.7 in terms of the formation volume factors and the solution gas-oil ratio (R_s). $\mathbf{u_m}$ is the phase velocity given by the Darcy's law

$$\mathbf{u_m} = -\beta \frac{\mathbf{K} k_{rm}}{\mu_m}(\nabla p_m - \gamma_m \nabla D), \qquad m \in \{o, w, g\} \tag{B.0.8}$$

where $\beta = 1.127 \times 10^{-3}$ for oil field unit system and 8.527×10^{-3} for metric unit system, $\mathbf{K}$ is the absolute permeability tensor assumed to be diagonal in this work, k_{rm} is the relative permeability of the phase, μ_m is the phase viscosity, p_m is the phase pressure, D is the true vertical depth between two points and γ_m is the specific weight of phase m given by

$$\gamma_m = \alpha_c \rho_m g,$$

where $\alpha_c = \frac{1}{144 g_c}$ for oil field unit system and 1 for metric system. g and g_c are the gravitational accelerations in metric and field units, respectively. The phase pressures are related by the capillary pressures given by

$$p_{cow} = p_o - p_w = f(S_w) \text{ and}$$
$$p_{cgw} = p_g - p_o = f(S_g).$$

In a multiphase system the relative permeability describes the flow of one phase in the presence of others. Three phase relative permeabilities can be found using the two phase data. Stone's model II is widely used to calculate the relative permeability of oil, given k_{rw} and k_{rg}, which is given by

$$k_{ro} = k_{rocw}\left[\left(\frac{k_{row(S_w)}}{k_{rocw}} + k_{rw}(Sw)\right)\left(\frac{k_{rog(S_g)}}{k_{rocw}} + k_{rg}(Sg)\right) - k_{rw}(S_w) + k_{rg}(S_g)\right].$$

Water phase properties area adapted from [21]. Oil and gas phase properties are evaluated based on input tables generated externally. For further information on the input data used in this research, please refer to Appendix D.

Finally, the well production rate for phase m completed in a gridcell n is calculated using

$$q_{m,n} = WI_n \left(\frac{k_{rm}}{\mu_m B_m}\right)_n [p_{o,n} - p_{wf,n}],$$

where WI is the well index given by

$$WI_n = \frac{\Delta Z \sqrt{(k_x k_y)_n}}{141.2 \left[\ln \frac{r_o}{r_w} + s_n\right]},$$

where s_n is the skin factor (assumed zero in this work) and $r_{o,n}$ is the equivalent radius given by

$$r_{o,n} = \frac{0.28103 \Delta x_n}{1 + \sqrt{(k_x k_y)_n}} \sqrt{1 + \left(\frac{k_x}{k_y}\right)_n \left(\frac{\Delta y}{\Delta x}\right)_n}.$$

The mass conservation equations (B.0.5, B.0.6 and B.0.7), assuming zero capillary pressure and neglecting gravity, can be rewritten in canonical forms given by

$$F_1(p_o, S_w, S_g) = \frac{\partial}{\partial t} a_1(p_o, S_w, S_g) - \nabla \cdot (\mathbf{G_1}(p_o, S_w, S_g) \nabla \Phi_o) - w_1(p_o, S_w, S_g), \quad \text{(B.0.9)}$$

$$F_2(p_o, S_w) = \frac{\partial}{\partial t} a_2(p_o, S_w) - \nabla \cdot (\mathbf{G_2}(p_o, S_w) \nabla \Phi_w) - w_2(p_o, S_w), \quad \text{(B.0.10)}$$

$$F_3(p_o, S_w, S_g) = \frac{\partial}{\partial t} a_3(p_o, S_w, S_g) - \nabla \cdot (\mathbf{G_3}(p_o, S_w, S_g) \nabla \Phi_g) - w_3(p_o, S_w, S_g), \quad \text{(B.0.11)}$$

where

$$a_1 = \phi(\rho_w S_w + \rho_o S_o + \rho_{go} S_o + \rho_g S_g), \qquad a_2 = \phi \rho_w S_w, \qquad a_3 = \phi(\rho_{go} S_o + \rho_g S_g), \tag{B.0.12}$$

$$\mathbf{G_1} = \rho_w \frac{\mathbf{K} k_{rw}}{\mu_w} + \rho_g \frac{\mathbf{K} k_{rg}}{\mu_g} + \rho_o (1 + R_s) \frac{\mathbf{K} k_{ro}}{\mu_o}, \tag{B.0.13}$$

$$\mathbf{G_2} = \rho_w \frac{\mathbf{K} k_{rw}}{\mu_w}, \tag{B.0.14}$$

$$\mathbf{G_3} = \rho_g \frac{\mathbf{K} k_{rg}}{\mu_g} + \rho_o R_s \frac{\mathbf{K} k_{ro}}{\mu_o}, \tag{B.0.15}$$

$$w_1 = q_t \qquad w_2 = q_w \qquad w_3 = q_g + R_s q_o. \tag{B.0.16}$$

Sequential implicit simulation treats either pressure or saturation fields as frozen while solving the mass conservation equations. Hence, in effect, $a_1(p_o, S_w, S_g) \equiv a_1(p_o)$, $a_2(p_o, S_w) \equiv a_2(S_w)$ and $a_1(p_o, S_w, S_g) \equiv a_3(S_g)$. In this work we decouple pressure from saturations which results in a decoupled elliptic equation and a system of hyperbolic equations. The first sequential step involves the solution of the pressure equation using frozen saturation states. Subsequently, a coupled system of saturation equations is solved assuming a frozen pressure field. The states are iteratively updated until the desired simulation time. The pseudo code for this sequential scheme is given in the following algorithm.

Algorithm 5: Three-phase sequential implicit scheme

Data: p_o^n , S_w^n, S_g^n

$p^\beta = p_o^n$, $S_w^\beta = S_w^n$, $S_g^\beta = S_g^n$;

do

  Solve $F_1(p_o^{\nu+1};\ S_w^\nu,\ s_g^\nu) = 0$;

  Solve $F_{2,3}(S_w^{\nu+1},\ S_g^{\nu+1};\ p_o^{\nu+1}) = 0$;

while $||(p_o^\beta - p_o^{\beta-1})|| \geq \epsilon_p$ *or* $||(S_w^\beta - S_w^{\beta-1})|| \geq \epsilon_s$ *or* $||(S_g^\beta - S_g^{\beta-1})|| \geq \epsilon_s$ *and* *MBAL ERROR* $\geq \epsilon_{mbal}$;

The tolerance values used in this work are $\epsilon_p = 0.5$, $\epsilon_s = 1.0E-04$ and $\epsilon_{mbal} = 1.0E-07$

APPENDIX C

NONLINEAR TRANSFORM RELATING LINEAR DIFFUSION-ADVECTION-REACTION PROBLEMS TO THE SCREENED POISSON EQUATION.

Consider a scalar field $u : D \subset \mathbb{R} \to \mathbb{R}$ that is twice differentiable on the open and bounded domain D, and that satisfies the linear PDE,

$$\begin{cases} \Delta u + \frac{\mathbf{b}}{a} \cdot \nabla u + \frac{c}{a} u = \frac{f}{a}, & x \in D \subset \mathbb{R}^3 \\ \nabla u . \hat{n} = 0 & x \in \partial D \end{cases}, \tag{C.0.17}$$

where $a : D \to \mathbb{R}^+$, $\mathbf{b} : D \to \mathbb{R}^3$ and $c : D \to \mathbb{R}^+$ are variable coefficients. We will characterize u by the solution to a certain screened Poisson equation given by

$$\Delta w - \lambda^2 w = g, \;\; x \in D \subset \mathbb{R}^3. \tag{C.0.18}$$

In particular, let $\alpha : D \to E \subset \mathbb{R}$ be a twice differentiable and bounded function (E is bounded), and consider the scalar field,

$$w \equiv e^{\alpha} u.$$

By direct application of the chain rule,

$$\nabla w = \nabla \alpha e^{\alpha} u + e^{\alpha} \nabla u \tag{C.0.19}$$

and

$$\Delta w = e^{\alpha}\Delta u + 2\nabla\alpha e^{\alpha}\nabla u + (\Delta\alpha e^{\alpha} + \nabla\alpha\cdot\nabla\alpha)u \tag{C.0.20}$$

Substituting Equations C.0.19 and C.0.20 into Equation C.0.18 and comparing with Equation C.0.17, we obtain

$$\alpha = \frac{1}{2}\int_{\Gamma[x_0,x]} \frac{\mathbf{b}}{a}dx, \tag{C.0.21}$$

$$\lambda = \sqrt{(\Delta\alpha + \nabla\alpha\cdot\nabla\alpha - c/a)} \quad \text{and} \tag{C.0.22}$$

$$g = \frac{f}{a}e^{\alpha}. \tag{C.0.23}$$

Similar transformation is applied to the boundary conditions. Subsequently, since by assumption α is bounded, w is solution of

$$\begin{cases} \Delta w - \lambda^2 w = g, & x \in D \subset \mathbb{R}^3 \\ (\nabla w - 0.5\nabla\alpha w)\cdot\hat{n} = 0 & x \in \partial D \end{cases}, \tag{C.0.24}$$

if and only if u is a solution to C.0.17.

APPENDIX D

NUMERICAL INPUT DATA FOR THE THREE-PHASE PROBLEM

This appendix covers the details of the numerical parameters and inputs used in this research.

D.1 Problem setup

Two and three dimensional problem settings are considered in this work with varying number of production and injection wells. In all the two dimensional problems without fractures there in 1 injector at the center and 4 producers on the periphery. Figure D.1 shows the well layout on the SPE10 comparative study case. In all the cases, wells are only completed in one layer. Permeability and porosity fields are case specific and are described

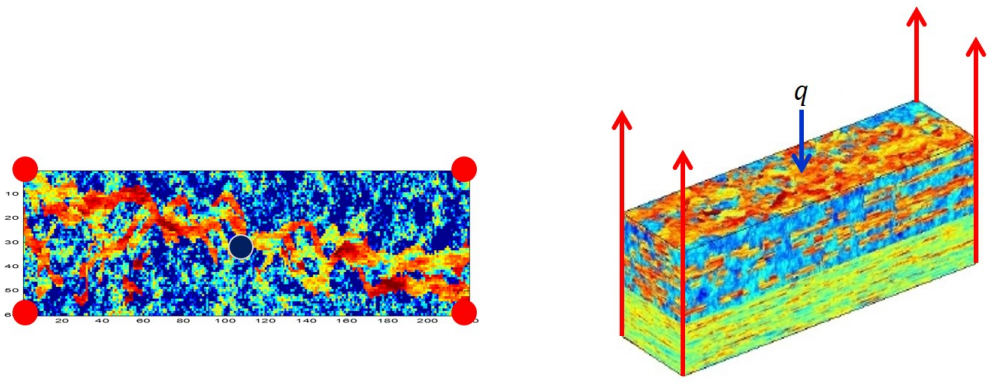


Figure D.1: Problem setting for a two and three dimensional case ([14])

in the text. The rock compressibility value is $3.4E-06$. The initial state parameters and well controls are tabulated below.

D.2 Rock properties

The relative permeabilities are calculated using the two phase data in the Stone's model II.

Phase	Oil-water-gas
Initial water saturation	0.25
Initial gas saturation	0.15
Initial reservoir pressure (psi)	2000.0
Water injection rate (bbl/day)	1500.0
Well bottom hole pressure (psi)	1550.0
Dimensions (ft x ft x ft)	1200.0 x 2200.0 x 15.0

Table D.1: Initial state information and well controls

S_g	K_{rg}	K_{rog}	P_{cog}
0.0	0.0	1.0	0.0
0.1	0.0	1.0	0.0
0.2	0.1	0.4	0.0
0.3	0.15	0.35	0.0
0.4	0.2	0.3	0.0
0.5	0.25	0.25	0.0
0.6	0.3	0.2	0.0
0.7	0.35	0.15	0.0
0.8	0.4	0.0	0.0
0.9	1.0	0.0	0.0
1.0	1.0	0.0	0.0

Table D.2: Gas-oil data

S_w	K_{rw}	K_{row}	P_{cow}
0.0	0.0	1.0	0.0
0.22	0.22	1.0	0.0
0.3	0.09	0.7	0.0
0.4	0.16	0.6	0.0
0.6	0.36	0.4	0.0
0.8	0.64	0.2	0.0
0.9	0.81	0.0	0.0
1.0	0.81	0.0	0.0

Table D.3: Water-oil data

D.3 Fluid properties

Correlations for the water phase properties are adapted from [21]. Input tables are used for oil and gas phase fluid properties. The reference densities of water, oil and gas are

P_o	R_s	B_o	μ_o
165	0.018	1.082	0.93
415	0.050	1.095	0.93
665	0.085	1.111	0.93
915	0.124	1.128	0.93
1165	0.165	1.146	0.93
1415	0.208	1.166	0.93
1665	0.252	1.187	0.93
1915	0.298	1.208	0.93
2165	0.345	1.231	0.93
2415	0.393	1.255	0.93
2502	0.420	1.263	0.93

Table D.4: Oil phase data

P_g	R_v	B_g	μ_g
300.0	0.0012	9.683	0.0234
600.0	0.0014	4.760	0.0234
900.0	0.0015	3.108	0.0234
1200.0	0.0019	2.285	0.0234
1500.0	0.0023	1.795	0.0234
1800.0	0.0027	1.475	0.0234
2100.0	0.0031	1.252	0.0234
2400.0	0.0033	1.090	0.0234
2700.0	0.0036	0.974	0.0234
3054.0	0.0039	0.866	0.0234

Table D.5: Gas phase data

$63.02 lb/ft^3$, $45.0 lb/ft^3$ and $0.0702 lb/ft^3$, respectively. The fluid compressibility value are $3.0E-06$, $3.0E-05$ and $1.0E-03$ for water, oil and gas, respectively.