SEQUENTIAL-IMPLICIT NEWTON'S METHOD FOR GEOTHERMAL RESERVOIR SIMULATION

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

Zhi Yang Wong

November 2018

This dissertation is online at: http://purl.stanford.edu/yd935db5070

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

Roland Horne, Co-Adviser

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

Hamdi Tchelepi, Co-Adviser

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

Pavel Tomin

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Numerical simulation of thermal multiphase fluid flow poses significant difficulties for nonlinear solvers. One approach to this problem is to solve the entire nonlinear system of equations simultaneously with a fully coupled method. However, due to the strong coupling and multiphysics interactions between equations, it is difficult to analyze and challenging to design solvers using this fully coupled method. Instead, a sequential-implicit method splits the multiphysics problem into different subproblems so that they can be each solved separately. The research described in this thesis developed and investigated sequential-implicit methods for geothermal simulation. The sequential-implicit method isolates each of the subproblems and enables the use of specialized solvers for each separate subproblem. Once each subproblem is solved in an efficient manner, the entire multiphysics problem is coupled through a sequential-implicit method. However, these sequential-implicit methods can face difficulties converging to the coupled solution. This thesis describes various sequential-implicit methods that reduce the computational cost of subsurface geothermal simulations by improving their nonlinear convergence. This split of the different physics allows for more specialized solvers such as multiscale finite volume or linear solver preconditioning methods to be built upon it.

We demonstrated that for sequential-implicit geothermal simulations, a hybrid constraint strategy is necessary. This hybrid approach involves imposing a fixed pressure constraint for single-phase cells (control volumes) and a fixed density for

two-phase cells. However, numerical comparisons showed that the outer loop convergence for the hybrid method on complex scenarios performed poorly in comparison to the fully coupled method. To improve on the outer loop convergence, a modified-sequential fully implicit method was investigated. Although the modified sequential-implicit method improves the outer loop convergence, the additional computational cost of the modified-sequential fully implicit method could diminish the gains from the improved convergence.

One of the main contributions of this work is the development of a sequential-implicit Newton's method. Sequential-implicit methods often suffer from slow convergence when there is a strong coupling between the individual subproblems. This is due to the slow linear convergence rate of the fixed-point iteration that is used in current sequential-implicit methods. This new sequential-implicit Newton's method follows the same sequential scheme as the current sequential-implicit fixed-point method, but with a faster quadratic convergence rate compared to the current linear convergence rate. This method is not only applicable to geothermal simulation but also to all sequential-implicit multiphysics simulations. We demonstrated the effectiveness of this approach on two different multiphysics porous media problems: flow-thermal in geothermal simulation and flow-mechanics in geomechanics reservoir simulation. The numerical experiments show an improvement in outer loop convergence across all multiphysics problems and test cases considered. For some specific cases where there was a strong coupling, up to two orders of magnitude improvement was seen in the outer loop convergence for the sequential-implicit Newton's method.

Following these investigations of the sequential-implicit method, we developed a sequential-implicit nonlinear solver to better solve a condensation problem in fully coupled geothermal simulation. This condensation problem is associated with the flow of cold water into a cell that is at saturated conditions that is also known as

a "negative compressibility" problem. In order to deal with this problem, the nonlinear solver must be modified. We developed a sequential-implicit nonlinear solution strategy that overcomes this nonlinear convergence problem associated with the condensation front. For one-dimensional problems, this nonlinear solution strategy converged for all timesteps sizes, while the fully coupled strategy only converged for a limited sized timestep. Furthermore, for a two-dimensional heterogeneous problem, the largest Courant-Friedrichs-Lewy (CFL) number for this method is at least an order of magnitude larger than the CFL number that can be used with the fully coupled approach.

Acknowledgments

I am deeply indebted to my two advisors, Prof. Roland Horne, and Prof. Hamdi Tchelepi. I am thankful for their guidance, encouragement, and support throughout my time at Stanford. I am also thankful for their infectious passion for research and our fruitful research discussions throughout my Ph.D. I feel so fortunate to have two extremely supportive advisors and for their continuous positive encouragement even when discussing unpromising research results. I am thankful for their mentorship and also for providing me with the independence required to develop as a researcher.

I would like to thank Dr. Pavel Tomin for agreeing to be part of my reading committee. I am also thankful for his guidance and deep understanding of the technical details in AD-GPRS. I am also thankful to Pavel for our friendship, deep discussions into the implementation and for always being willing to bounce ideas with. I would also like to thank Prof. Louis Durlofsky for serving on my defense committee as an oral examiner and Prof. Doug James for chairing my Ph.D. defense. His valuable comments are greatly appreciated.

I am also thankful to Dr. Ruslan Rin for his collaboration on the implementation of the General Sequential Framework, much of this research is a result of the elegant framework that he designed. I am thankful to Prof. Felix Kwok for his ideas and collaboration on the sequential-implicit Newton's method. I am grateful as well to Prof. Denis Voskov for his support, patience, and supervision during my Masters. I would also like to acknowledge Dr. Timur Garipov and Dr. Nicola Castelletto for

fruitful discussion and explanations about geomechanics and to Dr. Alexandre Lapene for discussions about negative compressibility. I would also like to acknowledge AD-GPRS team: Dr. Huanquan Pan, Dr. Michael Connolly, Kirill Terekhov, Sergey Klevtsov, Jiawei Li and Ruixiao Sun.

I also would like to express my gratitude to the Stanford Geothermal Program and the SUPRI-B research group for their friendship and research advice in group meetings. I am thankful for the Consortium on Reservoir Simulation Research at Stanford University (SUPRI-B) for supporting my doctoral studies.

I am thankful for all my friends in ERE and specifically for my office mates, Kyu, Markus, and Muhammad. My time in graduate school would have been extremely different without your laughter and camaraderie.

I am appreciative for the Stanford IVGrad group and my church RBF, for their friendship and community during my time here at Stanford. I am also grateful for my family and for their support and unconditional care while I study at Stanford. I am immensely grateful to Vivian for the love and support that she gives me. I would ultimately like to thank God for all the blessings he has provided for me in my life and especially this opportunity to study at Stanford.

Nomenclature

$\boldsymbol{\sigma}$	Total stress tensor
$\boldsymbol{u}$	Displacement vector
Δt	Time step
Δt_{comp}	Time step where pressure update for the first Newton step would diverge
Δt_{neg}	Time step where pressure update in full Newton is negative
ϵ^e	Second-order elasticity strain tensor
ϵ_F	Tolerance for flow residual
ϵ_T	Tolerance for thermal residual
κ_l	Thermal conduction transmissibility for the interface l
$\mathbb{C}$	Fourth-order tensor elasticity moduli tensor
$\mathcal{H}$	Function that computes the enthalpy solution based on the pressure and coupling variable c_F
$\mathcal{P}$	Function that computes the pressure solution based on the pressure and coupling variable c_T
μ_k	Viscosity of the phase k
ϕ	Porosity of the rock
ρ_k	Mass density of phase k
ρ_t	Total density
Υ	Transmissibility
Υ^l	Constant geometric part of the transmissibility

Υ_p^l	Transmissibility of the interface l for phase k
b	Second-order tensor of Biot coefficients
c_F	Coupling variable for flow equation
c_T	Coupling variable for thermal equation
C_{neg}	Timestep threshold for when pressure update will be negative
E	Young's Modulus
g	Gravitational constant
h	Total Enthalpy
h_k	Phase enthalpy of phase k
H_{inj}	Enthalpy of the injected fluid
J_F	Jacobian for the flow residual equation
J_T	Jacobian for the thermal residual equation
K	Total thermal conductivity of the fluids and rock
k	Rock permeability
k_{rk}	relative permeability for phase k
k_{rs}	Relative permeability of steam
k_{rw}	Relative permeability of liquid water
K_{SIFP}	Number of outer SIFP iterations
K_{SIN}	Number of outer SIN iterations
p	Pressure
p_k	Pressure of phase k
p_{in}	Constant pressure of the source term
Q	Source/sink term
R_F	Flow residual equation
R_T	Thermal residual equation
R_u	Residual form of the momentum balance for the mechanics equations
S_k	Saturation of phase k

T_{fact}	Time for factorization
T_{sub}	Time for substitution
U_k	Internal energy of phase k
u_k	Velocity of phase k
V	Cell volume
x_F^{n+1}	Primary variables for the flow equation
x_T^{n+1}	Primary variables for the thermal equation
z	Coordinate direction of gravity
Q_{Fi}	Source/sink mass contribution applied on cell i
Q_{Ti}	Source/sink energy contribution applied on cell i
AD-GPRS	Automatic-Differentiation General Purpose Research Simulator
FC	Fully Coupled
SIFP	Sequential-implicit Fixed Point
SIN	Sequential-implicit Newton

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Geothermal Simulation

The main focus of this research is to improve the computational efficiency of geothermal simulation by overcoming the nonlinear convergence issues faced by current algorithms. Geothermal energy is generated by harnessing hot steam or water from the ground to drive an electrical generator. This extraction process can be properly sustained through careful management of the production and reinjection of the fluid. Energy generated through this process results in a low amount greenhouse gas emission (50g CO_2/kWhe); this is four times less than solar PV and six times less than natural gas [63]. Geothermal energy can also act as the base load, as it is capable of producing energy at all times of the day. This makes geothermal energy a very attractive renewable energy source in comparison to other intermittent renewable sources. This is important for energy production as it balances out the energy fluctuations generated from intermittent renewable energy sources such as wind and solar. Although geothermal energy is globally sustainable, the energy extraction must be closely monitored to avoid local depletion so that it can maintain its economic and environmental sustainability.

An essential tool in the management and forecast of this depletion process is reservoir simulation [48]. These reservoir models are used to predict the performance of a geothermal field and to assess the outcomes of various production scenarios [2]. This ensures that there is a balance between the depletion of the heat and fluid in the reservoir and the recharge of heat and fluid back into the reservoir. This balance maintains the sustainability and economic viability of geothermal projects. The reliability of the reservoir models is strongly dependent on the complexity and accuracy of the physics representation.

To improve the reliability of geothermal reservoir simulations, uncertainty quantification is required to provide uncertainty estimates on the forecasts. Uncertainty quantification often requires an ensemble of different reservoir models to be simulated [6]. Utilizing an ensemble of models allows the modeler to incorporate the known uncertainty of the inputs and enables the reservoir simulator to produce results that reflect the input uncertainty. To optimize a geothermal project's performance, this requires an assessment of different scenarios that correspond to various reservoir models for each scenario. To run the ensemble of models for uncertainty quantification or optimization, it is necessary that the simulator is robust and efficient. This ensures that these models run in a realistic time frame while maintaining a low computational cost.

Geothermal reservoir simulation involves the numerical solution of the governing equations that model the fluid and heat transfer in the subsurface. These governing equations involve the coupled mass and energy balances that govern the movement of the fluid and energy in the subsurface. In addition, the fluid properties such as density, pressure, temperature, and enthalpy must follow their thermodynamic relationships. In the numerical simulation of geothermal reservoirs, one has to deal with thermal multiphase flow and transport. A review of geothermal reservoir models can be found in [48].

In the thermal displacement processes of interest, the energy balance is often tightly coupled with the mass conservation equations that describe the multiphase flow. One important source of this coupling is the thermodynamic relationships that each cell (control volume) has to satisfy. This nonlinear coupling of the conservation equations and constitutive relationships poses significant challenges to both nonlinear and linear solvers. An example of this issue is shown in [38] where they experienced difficulty when using a general purpose simulator TOUGH2 [54] to solve a discrete fracture model. This was thought to be due to the complexity of the numerical grid, thus resulting in a domain error. Another problem observed in TOUGH2 was seen by Noy et al. [46], where a stalling behavior in the time step was observed in a CO_2 sequestration simulation. For geothermal applications, this stalling behavior occurs typically in natural-state geothermal simulations, where large time steps are taken. Large time steps often challenge reservoir simulations, due either to inaccurate or ill-conditioned Jacobian matrices which are a result of the highly nonlinear nature of the governing equations.

In this research, we focused on the methods of simulation of pure-water geothermal reservoirs. While the phase behavior for the single-component (i.e., H_2O) case is easier to model than the cases with multicomponent hydrocarbon mixtures and water, the nonlinear coupling of the mass and energy conservation equations in pure-water systems is more pronounced. One important challenge in pure-water geothermal systems is that when the pore water undergoes boiling or condensation, the pressure and temperature become dependent variables. This has a direct impact on the mathematical formulation, space and time discretization schemes, and the solution strategy. Another challenge is that the contrasts in the density and viscosity of the liquid and vapor phases are large, and that leads to large differences in the phase mobilities.

1.2 Condensation Problem

One notable complexity due to the tight coupling between fluid flow and thermodynamic behavior occurs in a condensation problem. The condensation problem occurs when cold water is injected into a two-phase saturated cell. This results in the so-called "negative compressibility" problem [10]. This "negative compressibility" behavior occurs in cells that contain two fluid phases: water (liquid) and steam (vapor). This problem is most pronounced in pure-water systems, whereby depending on the pressure and temperature conditions, the single component (H_2O) may exist as a single-phase liquid, a single-phase vapor, or as a two-phase (liquid-vapor) system. Coats [10] described this phenomenon using a simple single-cell problem, where cold water is injected at a fixed pressure into a cell with saturated steam. As condensation occurs, the cell pressure decreases, and that enhances the inflow of cold water. Nevertheless, the cell pressure continues to decline as the steam condenses. As the last drops of steam condense, and liquid water, which has small compressibility, occupies the entire pore space enclosed by the cell, the cell pressure rises dramatically. The cell pressure ultimately rises to the injection pressure, and the inflow of water ceases. This behavior can be explained as follows. During the condensation process, the reduction in the vapor-phase volume overwhelms the expansion of the fluids due to compressibility effects, and the over-pressure displays this "negative compressibility" behavior.

This "negative compressibility" effect has also been discussed in more detail by Pruess et al. [53] and Falta et al. [14]. In those analyses, the "negative compressibility" phenomenon was attributed to the idealization that there is complete thermodynamic equilibrium combined with a computational mesh (grid) of finite size. It was found that this idealization results in spurious pressure variation in the cells that contain the two-phase front.

These pressure oscillations present a severe limitation for the nonlinear convergence of these problems [21]. The solution presented by Gudbjerg et al. [21] was to artificially restrict any flow of cold water into the blocks that are steam saturated. For their specific flow regime where steam is injected into cold water, Gudbjerg et al.ś [21] approach circumvents the "negative compressibility" issue by preventing the nonphysical back-flow of cold water. However, for the case where water is being injected into steam with a large temperature difference, their approach would produce nonphysical results. This limits the application of their solution in most geothermal fields where there are often large temperature differences, such as the case where cold water is reinjected back into the reservoir.

Wang [73] presented an analysis of the "negative compressibility" issue for fully coupled formulations. In that analysis, a stability criterion for the timestep was developed to ensure convergence of the fully implicit solution, thus preventing unnecessary timestep cuts. However, the derived stability criterion enforces a severe limitation on the allowable timestep size.

The main issue with the current literature on the condensation problem is that it is all based on a fully coupled formulation. This involves the simultaneous solution of the entire multiphysics problem. This makes the problem difficult to analyze and understand. If instead, we are able to split the problem into the key complexities and solve each complexity separately, this would provide a starting point of understanding how to tackle the phenomena. Once this condensation problem is understood, this could greatly increase the capability of geothermal simulation. Overcoming this specific nonlinear convergence problem would decrease the computational time and improve the robustness of geothermal simulations. This would, in turn, improve the forecasting capability of geothermal reservoir models and sustainability of geothermal energy.

1.3 Sequential-implicit Simulation

To address this strong coupling, a common approach [54, 80] in numerical simulation is to use a fully coupled (FC) and fully implicit method. Although the FC ensures numerical stability in the problem, it does not guarantee nonlinear convergence. In addition, both of the mass and energy conservation equations have parabolic and hyperbolic behavior. The system of equations is parabolic in the flow and conduction and hyperbolic in the transport of mass and energy. Coupling the different physics and flow mechanisms makes it difficult to investigate this nonlinear problem. A separation of the flow and thermal equations could reduce the severity of the nonlinear coupling difficulties and improve the current understanding of the problem. Depending on the coupling strength of the problem, separating these equations could also reduce the overall computational time, because this reduces the number of linear equations solved. However, decoupling could also increase the number of iterations required to couple the separate problems when the coupling between them is too strong.

Sequential methods have proven to be attractive when solving multiphysics problems involving flow and mechanics. Solving flow and mechanics problems with a fully coupled approach have shown to have significant linear solver scalability issues [74, 75, 7, 31]. To tackle these issues, Klevtsov et al. [31] designed a block preconditioned Krylov method for multiphase flow and geomechanics. This block preconditioner was applied in a two-stage approach, where the pressure and displacements are decoupled in the first stage. Castelletto et al. [7] showed that the block preconditioning step can also be interpreted as the fixed stress sequential approach. Here we see that even in the design of the fully coupled linear solution, it is inspired by and relies on the understanding of the sequential approach.

A sequential strategy reduces the complexities and requirements of the linear solver, and this makes them easily scalable in comparison to a fully coupled approach.

A sequential coupling in flow and mechanics can also be beneficial when they have different computational domain sizes (geomechanics domain is often much larger than flow), different spatial discretization schemes (finite volume for flow and finite element for mechanics) or different simulators for flow and mechanics (e.g. TOUGH2 [54] for flow and ABAQUS for mechanics [56]). To improve these sequential methods, the coupling between the flow and mechanics problem has been investigated extensively. It was shown that the fixed-stress and undrained split are unconditionally stable and that the fixed stress converges faster than undrained split [28, 30, 29, 40]. However, the performance of the fixed stress method is strongly dependent on the coupling strength between the flow and mechanics problems [28]. If the coupling strength of the problem is too high, this could result in a slow convergence rate thus requiring too many sequential iterations to couple the two problems together. This coupling strength is dependent on the input rock properties of the problem. So depending on the reservoir that is investigated, this could greatly limit the capability of these sequential-implicit methods for strongly coupled problems.

When modeling isothermal multiphase flow and transport in porous media, a sequential strategy has also proven to be useful. It was first proposed in a Multiscale Finite Volume formulation [26] to simulate immiscible multiphase flow in porous media. A sequential solution strategy was essential to the multiscale formulation where the near-elliptic flow and hyperbolic transport problems are solved separately. The separation of the elliptic and hyperbolic parts allowed for these advanced discretization schemes and scalable algorithms to be utilized. This is essential for multiscale methods as it relies on solving only the elliptic part of the flow problem, [25]. On the linear solver level, this split of the elliptic aspect is the fundamental idea of the widely used constrained pressure residual (CPR) preconditioner [71]. The CPR preconditioner is a two-stage preconditioner that first isolates the near-elliptic part which is solved with an efficient elliptic solver such as a multigrid solver. The rest of the

unknowns are then solved with a local smoother such as ILU(k). This two-stage split is again motivated by the physical understanding of the elliptic and hyperbolic parts of the problem.

Similar to the flow and mechanics problem, a strong coupling between the flow and transport problems would also require a large number of outer iterations. A large number of outer iterations would overshadow the computational gains from using these specialized solvers. There has been a significant amount of recent work in trying to improve this separation to decrease the number of outer iterations [33, 44, 45, 42, 43]. Their approaches involved modifying the equations of each subproblem to reduce the splitting errors, thus reducing the number of sequential iterations. It was found that the specific details of how each physical problem is linearized, split and formulated have strong contributions to the splitting-error of the sequential scheme [43] and consequently the number of sequential iterations. To reduce this splitting error, Moncorgé et al. [42] used the phase appearance and disappearance of the grid cells to determine how to enrich the pressure equation.

A sequential strategy was also essential in the analysis of the nonlinearities in flow and transport problems. A common approach for this analysis is to separate the equations into pressure and saturation equations, this allows the hyperbolic saturation equation to be investigated separately. It was shown that the flux function was the key to understanding nonlinearities for nonlinear flow and transport problems. Jenny et al. [27] developed a nonlinear solver based on the saturation equation for isothermal two-phase flow. They used the analytical flux function to determine the inflection point and thus guide their Newton solver for saturation updates. Wang and Tchelepi[72] extended this approach to two-phase problems with viscous and buoyancy forces by using a 'trust-region' type solver to determine under which saturation values the Newton's method was guaranteed to converge. These intervals were based upon the inflection points and unit flux points of the analytical flux function. Li

and Tchelepi [34] improved upon the analysis of Wang [72] to show that these trust regions should be computed for the numerical and not for the analytical flux. Hamon and Tchelepi [24] conducted an analysis of an implicit hybrid-upwinding strategy and the influence of this method on the flux function. These nonlinear analyses conducted for transport problems relies heavily on the sequential split of the flow and transport parts of the problem.

The available literature for similar types of analysis is less extensive for geothermal simulations. This is because the flow and thermal equations are often both solved with a finite volume scheme thus making the fully coupled formulation an appealing choice. There has been work to separate the parabolic and hyperbolic parts of the flow and thermal equations [65, 66]. [65] and [66] analyzed the sequential method for thermal problems. The focus of their formulations was to demonstrate the ability for pressure to be solved implicitly followed by an explicit solution for enthalpy. This is different from a sequential fully implicit method, where both pressure and enthalpy are solved implicitly and iteratively coupled. Due to the explicitness of enthalpy, all the numerical tests were conducted for a CFL < 1. Here the CFL is defined as:

$$\text{CFL} = \frac{Q\Delta t}{PV} \tag{1.1}$$

where: Δt is the time step, Q is the throughput in a cell and PV is the pore volume. The CFL is a normalized timestep and measures how the large the timestep is relative to the flow through a cell. Shown in [66] CFL is relatively low in comparison to the CFL size obtainable for complex sequential fully implicit problems in flow and transport problems [27, 72, 34, 23, 24]. This CFL also makes the applications of these algorithms infeasible for practical geothermal simulations especially natural state simulations, where large timesteps are desired for efficient simulations [49].

Two key themes emerge from the survey of sequential-implicit methods for porous

media problems. The first is that there are very limited studies for sequential-implicit methods in geothermal problems compared to other multiphysics problems. A logical first step would be to conduct similar types of analysis for geothermal simulation. This analysis would be guided and inspired by the work for flow-mechanics and flow-transport simulations. Findings from geothermal simulations could also further improve flow-mechanics and flow-transport simulations too. The second theme is that the main bottleneck from these sequential formulations is the slow outer loop convergence for strongly coupled simulations. For flow-mechanics, we see that the performance of these sequential simulations is greatly dependent on the coupling strength of the problem. This is problematic as the coupling strength of a problem is dependent on the input rock and fluid parameters. As this is dependent on the input rock properties, this would render certain reservoirs inappropriate for sequential simulation due to their coupling strength [55]. For flow-transport problems, even if highly specialized and efficient solvers are designed for the pressure and saturation equations, their practical applicability is now limited by the sequential coupling between them [32]. Therefore, designing a sequential-implicit algorithm that requires a small number of outer loops for convergence will be vital for practical applications.

The sequential-implicit method could be the key to understanding and improve the nonlinear convergence for geothermal simulations. If the nonlinear convergence is improved, this would greatly enhance the efficiency and robustness of geothermal simulators. We see that not only does this affect the understanding on the nonlinear level, but this could also shed light on designing efficient physics-based preconditioners for fully coupled linear solvers [71, 7]. Decreasing the computational time to run these simulations would allow users of the simulation tool to run more forecasts and have a better understanding of the reservoir model. This would thus improve the economic performance and sustainability of geothermal reservoirs.

1.4 AD-GPRS

A suitable platform for these investigations on the sequential-implicit methods for geothermal simulation is the Automatic-Differentiation General Purpose Research Simulator (AD-GPRS) [68, 83, 85]. AD-GPRS provides a general implicit coupling framework for solving multiphysics problems, [55]. The framework employs a modular design for each physics and allows the development and testing of different coupling strategies within a unified interface. Utilizing AD-GPRS greatly reduces efforts to implement each strategy and ensures a consistent performance comparison between different coupling schemes. The work described in this dissertation integrated the geothermal module [76, 77] into the new AD-GPRS framework.

In addition to the general implicit coupling framework, AD-GPRS has a wide range of capabilities in advanced physical modeling and numerical methods. These capabilities include thermal-compositional, EOS-based, multiphase flow and transport models [80, 69], generalized nonlinear formulations [68], multistage linear solvers [84, 31], complex multisegment wells [83], and nonlinear mechanical deformation models [19, 20].

The most important aspect in AD-GPRS for this work is the General Implicit Coupling Framework (GENIC) [55]. GENIC provides a framework where various coupling strategies for complex multiphysics simulation could be quickly prototyped and investigated. It was designed to be flexible and extensible, allowing the user to experiment and design a solution strategy for different multiphysics processes. The GENIC platform was designed on three key principles: modular design, physics extension, and seamless coupling. The framework employs a modular code design by splitting the individual physics into a set of different subproblems. The main components of this framework use a subproblem tree structure and abstract computational domains to separate and organize the variable sets for each subproblem. This modularity and

extensibility allow for quick prototyping of a wide range of solution strategies. The Automatic-Differentiation Expression Template Library (ADETL) [79] library is used to the full potential to allow for this seamless coupling. ADETL provides a generic representation of simulation variables to allow for automatic calculation of sparse Jacobians. This is vital to the investigations of different coupling strategies that result in different Jacobian structures. [82, 55] extended the work by [79] to allow for statuses to represent different primary variable sets with a flexible choice of residual equations. This allows for a flexible definition of the primary variables and residual equations that are solved for at each solution step.

GENIC provides a sequential coupling framework to easily prototype different sequential solution strategies for solving geothermal simulation problems. It allows for minimal code duplication and a consistent framework to test different sequential strategies. This enables one to isolate any computational gains to be determined solely on the algorithm implemented rather than being attributed to implementation details. This is pertinent to geothermal simulation, where because of the complexities of these algorithms, even with very similar algorithmic details, there can be large discrepancies in robustness and nonlinear convergence [12].

Utilizing AD-GPRS, various formulation and algorithmic comparisons in AD-GPRS have been conducted [68, 81, 77]. Voskov and Tchelepi [68] conducted a comparison between different widely used nonlinear formulations for general-purpose compositional reservoir simulation. It was found that a natural variable formulation approach performed the best for isothermal immiscible gas injection processes. It was found that for cases where a large number of phase changes occurred, formulations with the saturation as the primary variable would provide the best guide to the nonlinear solver. In the later update [81], it was demonstrated that if the gas injection performs at miscible conditions, the molar formulation performs much better due to the formal disappearance of the two-phase region. Wong et al. [77]

conducted a similar study for a geothermal simulation where the natural variable formulation was compared with a pressure-enthalpy formulation. The key difference between the two variable formulations was that the natural variable formulation (pressure-temperature-saturation) involved a variable switching process depending on the phase state, while the pressure-enthalpy formulation remained consistent regardless of the phase state. A similar model dependence was found when comparing the two primary variable formulations.

The sequential-implicit method questions a fundamental solution strategy idea for geothermal simulation that the flow and thermal equations should be solved separately rather than simultaneously in a fully coupled manner. Traditionally to investigate this idea and related algorithms would require the construction of a completely new simulator code. Not only is this tedious and time consuming, but any implementation issues could have a significant effect on the algorithm's performance, rendering the analysis obsolete. A consistent framework allows for the analysis and algorithmic gains to be transferable to all geothermal simulators. Implementing these sequential solution strategies with the AD-GPRS framework would also allow for this work to also be extended to another multiphysics coupling (e.g. mechanics or reactions) or numerical discretization schemes (e.g. Multipoint Flux Approximation). The goal for the results produced by exploiting this framework is essentially to improve the understanding of the nonlinear convergence and robustness of geothermal simulation.

1.5 Dissertation Outline

The main motivation of this research was to improve the computational efficiency of geothermal simulation by overcoming the nonlinear convergence issues faced by current algorithms. Specifically, the sequential-implicit method was investigated to both

provide computational speed-up and to better understand the nonlinearities demonstrated in these problems. This dissertation is split into three main chapters. These three chapters are organized to highlight the progression of work in investigating the sequential-implicit method for geothermal simulation.

1. **Sequential-implicit Method for Geothermal Simulation**

   This first chapter focuses on investigating the sequential-implicit method for geothermal simulation. The ideas and methods for this chapter were inspired by the existing work available in flow-mechanics and flow-transport problems. The first aspect investigated was the constraints applied to flow and thermal subproblems in a sequential-implicit strategy. This is analogous to the work done in [28] where the fixed stress scheme was found to perform the best out of all sequential strategies. In this work, we found that the hybrid method, a fixed pressure for single-phase and fixed-density for two-phase performed the best for geothermal simulations. However, the fully coupled method still outperformed all the sequential strategies.

   To improve upon this, a modified-sequential fully implicit method was investigated for geothermal simulations. This was motivated by the work done by Moncorgé et al. [42]. This was designed to overcome the poor convergence of the standard sequential method found earlier in the chapter. This improved the convergence of the outer loop for the sequential formulation; however, these additional equations increased the cost of each sequential iteration which could negate the benefits of solving in a sequential manner.

2. **Sequential-implicit Newton's Method**

   One of the key conclusions of the first chapter was that due to the strong coupling between the flow and thermal equations, the sequential-implicit method

would often perform poorly due to a large number of outer loop sequential iterations. This second chapter presents a sequential-implicit Newton's method that aims to address this issue. The sequential-implicit Newton's method presented is a general method that can not only be applied to geothermal simulations but also to any sequential-implicit multiphysics simulation. The key idea for this method is that these sequential-implicit methods could be interpreted as a fixed-point iteration approach. To improve on the current fixed-point iteration approach, a Newton's method update is presented. This improves the outer loop sequential convergence rate to have quadratic convergence rather than linear convergence. This sequential-implicit Newton's method follows the same sequential scheme as the fixed-point method, but with a faster outer loop convergence rate. This allows for both the ability to split the flow and thermal problem and a quadratic convergence rate of Newton's method. To demonstrate the generality of the sequential-implicit Newton's method to other multiphysics problems, a numerical comparison study was also conducted on a flow and mechanics problem.

3. **Sequential-implicit Nonlinear Solver for the Condensation Problem**

   In this third chapter, we describe a nonlinear preconditioner that was developed for the fully coupled method that is most commonly used in the literature. This preconditioner involves a sequential-implicit strategy and targets specifically the condensation problem. The idea of this preconditioner stemmed from an understanding of the sequential-implicit strategy on a single-cell problem. The preconditioner avoids the severe timestep limitation on this problem, or unphysical solutions, as presented in [21] and [73]. The preconditioner is based on a sequential fully implicit approach that provides a good initial guess to Newton's Method. This initial guess avoids the issues related to the "negative

compressibility" problem. The work presented in this chapter is a convincing demonstration of how the sequential-implicit method sheds light on difficult nonlinear convergence issues such as the condensation problem described.

Chapter 2

Sequential-Implicit Method for Geothermal Simulation

2.1 Introduction

A sequential-implicit method involves splitting the entire system of nonlinear equations into parts and solving each of them sequentially. This is in contrast to the commonly used method known as the fully coupled fully implicit method, where the entire system of nonlinear equations is solved simultaneously within a single Newton's method. The fully coupled method is desirable as it is unconditionally stable, but this comes at a high computational cost for a low-order approximation. If a sequential-implicit method is stable and convergent, it will be identical to the fully coupled method.

There are two common reasons to use a sequential-implicit method. The first is the case where there are two existing separate simulators or solvers and using these two existing simulators in a fully coupled way would be too difficult or time-consuming to implement [56]. The second is to provide a source of computational speed up by solving the two problems sequentially. The underlying source of computational

gain for a sequential-implicit method is that it allows one to divide the problem into smaller and easier to solve subproblems and solving many of these small subproblems would be faster than solving them all at once. These smaller subproblems are easier to solve in a few ways. The smaller subproblems would have fewer equations to solve at each step, this reduces the computational complexity of the linear solver. Also, they would likely all exhibit the same characteristics (elliptic, hyperbolic), thus making it easier to design linear solvers to target the specific characteristic of the equation [31]. Splitting the system into the separate parts would also make it easier to analyze and understand the nonlinearities associated with each subproblem [27, 72, 34]. Analysis from the sequential scheme was also the foundation for the structure of multistage preconditioners for fully coupled linear systems [7]. Similarly, understanding sequential-implicit methods for geothermal reservoir simulation would provide an opportunity for these types of analyses and computational gains.

When solving each individual subproblem, constraints are often applied to the variable set that is not solved. A naive implementation of simply fixing the other variable set that is not solved is likely to be unstable or divergent [28, 30, 29]. [28] investigated four different constraints and coupling strategies. A drained split, undrained split, fixed strain split and a fixed stress split. Kim et al. [28] showed that the fixed strain and drained split are conditionally stable and face severe stability issues when undergoing a plastic regime. It was also shown that the fixed stress and undrained split were both unconditionally stable and for simulation, at a fixed number of iterations, the fixed stress scheme is more accurate. The results demonstrated in [28] were supported by rigorous analysis in [30] and [29] where they showed that the fixed stress scheme is contractive and B-stable for the fully implicit backward Euler method (B-stable for $\alpha \geq 0.5$, where α is the parameter of time discretization of the midpoint rule $t_{n+\alpha}$, $\alpha = 1$ for backward Euler). The results presented from these studies had an immediate and widespread applicability for flow and mechanics simulation.

Currently, one of the key issues faced by sequential-implicit schemes is slow convergence when coupling the individual subproblems together. This slow convergence could negate any of the benefits of the sequential method. To tackle this issue in isothermal multiphase flow and transport, Moncorgé et al. [42] analyzed the nonlinear coupling between the flow and transport equations. From their analysis, they developed a modified sequential fully implicit (m-SFI) scheme that enriched the pressure equation with the coupling from saturation and compositional unknowns. This m-SFI method improved the robustness and convergence of the standard sequential-implicit scheme and is as robust with a similar convergence rate to the fully coupled method.

The work described in this chapter was motivated by these two works discussed ([28, 30, 29, 42]). The work consisted of two main investigations. The first was to investigate different sequential constraints for geothermal reservoir simulation. The focus of these investigations was focused on comparing the nonlinear performance of these different sequential-implicit constraints for different numerical examples. AD-GPRS [55] provided a consistent framework to run comparisons between these sequential-implicit constraints and the fully coupled method. The numerical models tested involved varying levels of complexities. The second investigation described in this chapter involved an m-SFI formulation for geothermal simulation. Here we redesigned the m-SFI method presented by Moncorgé et al. [42] for geothermal reservoir simulations. This part of the investigation focused on how the convergence of sequential methods could be improved so that it is more comparable to the fully coupled method in a geothermal simulation. Computational gains from a sequential-implicit method would improve the speed of geothermal simulations and thus the ability to manage the sustainability of geothermal energy. The investigations of a sequential-implicit scheme would also provide the foundation for understanding the complexities and nonlinearities of geothermal simulation for future improvements.

2.2 Conservation Equations

2.2.1 Governing Equations

In this chapter the flow and thermal residual equations are the mass and energy conservation equations for a single-component water that can exist in two fluid phases:

$$\frac{\partial}{\partial t}\left(\phi \sum_{k=1}^{2} \rho_k S_k\right) - \nabla.\left(\sum_{k=1}^{2} (\rho_k u_k)\right) - Q_M = 0 \tag{2.1}$$

and:

$$\frac{\partial}{\partial t}\left[(1-\phi)\,\rho_R U_R + \phi \sum_{k=1}^{2} \rho_k U_k S_k\right] - \nabla.\left(\sum_{k=1}^{2} (\rho_k h_k u_k)\right) - \nabla.\,(K \nabla T) - Q_E = 0 \tag{2.2}$$

where:

- ϕ is the porosity of the rock
- ρ_k is the mass density of phase k
- S_k is the saturation of phase k
- u_k is the velocity of the phase k
- Q is the source/sink term
- h_k is the phase enthalpy of phase k
- U_k is the internal energy of phase k
- K is the total thermal conductivity of the fluids and rock.
- The subscripts k represents the phase of the fluid, R the rock

ρ_k, h_k of each phase depend on the phase state of the fluid. For single-phase conditions, ρ_k and h_k are functions of pressure and temperature. The thermodynamic relationships for water are taken from [15]. However, for two-phase conditions, ρ_k and h_k will depend only on the pressure because now pressure and temperature are dependent ($p = p_{sat}(T)$). In addition to these two conservation equations, the saturation constraint must be satisfied; that is, the sum of all the phase saturation is unity.

$$\sum_{k=1}^{2} S_k = 1 \tag{2.3}$$

Darcy's Law

To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium:

$$u_k = -\frac{k k_{rk}}{\mu_k} \nabla (p_k + \rho_k g z) \tag{2.4}$$

where:

- u_k is the velocity of the phase k
- k is the rock permeability
- k_{rk} is the relative permeability for phase k
- p_k is the pressure of phase k
- g is the gravitational constant
- μ_k is the viscosity of the phase k
- z is the coordinate direction of gravity

2.2.2 Residual Form

The fully implicit finite volume approximation results a system of equations involving the flow and thermal residual equations:

$$\begin{aligned} R_F^{n+1}(x_F^{n+1}, x_T^{n+1})_i =& V\left(\left[\phi\sum_k^2 \rho_k S_k\right]_i^{n+1} - \left[\phi\sum_k^2 \rho_k S_k\right]_i^{n}\right) \\ & - \Delta t \sum_l . \left(\sum_{k=1}^2 \left(\rho_k \Upsilon_k^l \Delta\Phi_k\right)\right)^{n+1} - {Q_F}_i^{n+1}\Delta t = 0 \end{aligned} \tag{2.5}$$

$$\begin{aligned} R_T^{n+1}(x_F^{n+1}, x_T^{n+1})_i = V & \left(\left[(1-\phi)\,\rho_R U_R + \phi\sum_{k=1}^2 \rho_k U_k S_k\right]_i^{n+1}\right. \\ & \left. - \left[(1-\phi)\,\rho_R U_R + \phi\sum_{k=1}^2 \rho_k U_k S_k\right]_i^{n}\right) \\ & - \Delta t \sum_l \left(\sum_{k=1}^2 \left(\rho_k h_k \Upsilon_k^l \Delta\Phi_k\right)\right) - \Delta t \sum_l \left(\kappa_l (T_i - T_j)\right) - {Q_T}_i \Delta t = 0 \end{aligned} \tag{2.6}$$

where:

- l the upper index represents all quantities defined at the interface between connected cells
- $\Upsilon_p^l = \Upsilon^l \frac{k_{rk}}{\mu_k^l}$ is the transmissibility of the interface l for phase k
- Υ^l is the constant geometric part of the transmissibility
- κ_l is the thermal conduction transmissibility for the interface l,
- x_F^{n+1}, x_T^{n+1} represents the primary variables for the flow and thermal equations at the time level $(n+1)$.

- i is the index of the cell for this residual equation
- Δt is the time step
- $V = V(x, y, z)$ is the cell volume where j is a neighbor of cell i
- Q_{F_i} is the source/sink mass contribution applied on cell i
- Q_{T_i} is the source/sink energy contribution applied on cell i

2.3 Sequential Fully Implicit Method

2.3.1 Fully Coupled Method

The fully coupled method solves all the residual equations simultaneously:

$$\begin{aligned} R_F^{n+1}(x_F^{n+1}, x_T^{n+1}) &= 0 \\ R_T^{n+1}(x_F^{n+1}, x_T^{n+1}) &= 0 \end{aligned} \tag{2.7}$$

Using Newton's method to solve this system of nonlinear equations this leads to:

$$\begin{bmatrix} \frac{\partial R_F}{\partial x_F} & \frac{\partial R_F}{\partial x_T} \\ \frac{\partial R_F}{\partial x_F} & \frac{\partial R_T}{\partial x_T} \end{bmatrix} \begin{bmatrix} \delta x_F \\ \delta x_T \end{bmatrix} = - \begin{bmatrix} R_F \\ R_T \end{bmatrix} \tag{2.8}$$

For this study, we only examined using a pressure-enthalpy formulation as described in [77] ($x_F := p, x_T := h$).

2.3.2 Sequential-Implicit Method

For the sequential-implicit method, the flow and thermal equations are solved separately for the respective primary variables x_F, x_T and a constraint is added to each

equation to constrain the variable set that is not solved. The pressure-enthalpy formulation does not require any variable switching and thus allows for a convenient implementation and analysis of the sequential-implicit method. It is important to note that the sequential-implicit method is coupled iteratively until both the flow and thermal equation are below a tolerance. If the solution strategy is stable, the converged solution for the fully coupled and sequential-implicit will be identical up to the desired tolerance.

Algorithm 2.1 Sequential Fully Implicit

$p^{\nu^*} = p^n, h^{\eta^*} = h^n$

while $|||[R_F; R_T]||_2 > \epsilon_R$ **do**

$c_F^{\nu^*} = c_F(p^{\nu^*}, h^{\eta^*})$

while $||R_F||_2 > \epsilon_F$ **do** (Step 1)

$dp = -J_F(p^\nu, \mathcal{H}(p^\nu, c_F^{\eta^*}))^{-1} R_F(p^\nu, \mathcal{H}(p, c_F^{\nu^*}))$

$p^{\nu+1} = p^\nu + dp$

end while

$p^{\nu^*} = p^{\nu+1}$

$c_T^{\nu^*} = c_T(p^{\nu^*}, c_F^{\eta^*})$

while $||R_T||_2 > \epsilon_T$ **do** (Step 2)

$dh = -J_T^{-1}(\mathcal{P}(c_T^{\nu^*}, h^\eta), h^\eta), R_T(\mathcal{P}(c_T^{\nu^*}, h^\eta), h^\eta)$

$h^{\eta+1} = h^\eta + dh$

end while

$h^{\eta^*} = h^\eta$

end while

Where:

- p and h are the pressure and enthalpy solution respectively
- ν is the index for the Newton iteration in the flow subloop, and ν^* is the index

for the converged solution for the flow residual equation

- η is the index for the Newton iteration in the thermal subloop, and η^* is the index for the converged solution for the thermal residual equation
- J_F is the Jacobian for the flow residual equation (R_F) that is a function of the pressure and enthalpy solutions
- J_T is the Jacobian for the thermal residual equation (R_T) that is a function of the pressure and enthalpy solutions
- ϵ_R is the convergence criterion for both the flow and thermal residuals
- ϵ_F is the convergence criterion for the flow residual
- ϵ_T is the convergence criterion for the thermal residual
- c_F and c_T are the coupling variables for the flow and thermal equations
- $\mathcal{H}$ is a function that computes the enthalpy solution based on the pressure and c_F variable for each cell
- $\mathcal{P}$ is a function that computes the pressure solution based on the enthalpy and c_T variable for each cell

We will first describe the sequential fully implicit method for flow and thermal with general coupling terms and then describe the different coupling terms. Algorithm 2.1 shows an overview of the entire sequential-implicit method.

Step 1: Solve:

$$R_F^{n+1}(p^{n+1}, \mathcal{H}(p^{\nu^*}, \partial c_F = 0)) = 0 \tag{2.9}$$

This involves solving the mass residual equation assuming that the coupling variable c_F (e.g. $c_F := h, \rho, ...$) remains fixed.

Step 2: Solve:

$$R_T^{n+1}(\mathcal{P}(\partial c_T = 0, h^{n+1}), h^{n+1}) = 0 \tag{2.10}$$

This involves solving the energy residual equation assuming that the coupling variable c_T (e.g. $c_T := p, \rho, ...$) remains fixed. Once the thermal equation is solved with Newton's method, Step 1 (Equation) and 2 (Equation 2.10) are repeated until convergence.

In this study, we presented investigations for three different constraints: fixed pressure, fixed density and a hybrid approach. All three constraints are applied in the second step when the thermal residual is solved. A constant enthalpy constraint was applied for all three of the schemes described.

Fixed Pressure

The fixed pressure scheme assumes that the pressure at each cell is fixed when solving for the thermal residual $c := p$. So $\partial p = 0$:

$$R_T(p^{\nu^*}, h^{\eta+1}) = 0 \tag{2.11}$$

$$p^{\eta} - p^{\nu^*} = 0 \tag{2.12}$$

Fixed Density

The fixed density scheme assumes that the density at each cell is fixed when solving for the thermal residual $c := \rho$. So $\rho^* = \rho(p^{\nu^*}, h^k)$, where ρ is a function that computes the cell center densities based on the pressure and enthalpy of each cell. ($\rho : \mathbb{R}^{2N_c} \to \mathbb{R}^{N_c}$)

$$R_T(\mathcal{P}(p^{\nu^*}, \eta), h^{\eta}) = 0 \tag{2.13}$$

This means the below equation is satisfied at each Newton step in the thermal subloop:

$$\rho(p^{\eta}, h^{\eta}) - \rho^{\nu^*} = 0 \tag{2.14}$$

Fixed Pressure and Fixed Density (Hybrid)

The hybrid scheme combines the fixed pressure and fixed density constraints. This constraint varies cell-wise based on the phase state of the cell. A fixed pressure is enforced for single-phase cells and fixed density for two-phase cells. We determine the phase state of the cell by comparing the enthalpy with the saturated enthalpy of the cell, $h_w(p)$ and $h_s(p)$ are the saturated water and steam enthalpy respectively. The constraint for each cell i with pressure p_i and and enthalpy h_i:

$$c(p_i, h_i) := \begin{cases} \rho(p_i, h_i) & h_w(p_i) \leq h_i \leq h_s(p_i) \text{ (Two-phase)} \\ p_i & \text{otherwise (Single-phase)} \end{cases} \tag{2.15}$$

2.4 AD-GPRS

2.4.1 Sequential Framework

AD-GPRS is built on the general implicit coupling framework (GENIC). Using this coupling framework, each individual type of physics is split into different subproblems. Here for geothermal simulation, we have two types of subproblems a FLOW and THERMAL subproblem. To specify the type of coupling into AD-GPRS, we use the COUPLING keyword. An example of this is shown in AD-GPRS Input 2.1. In this definition, we also specify the formulation of the flow as a pressure (MOLAR_GEO) and enthalpy for the thermal residual equation (ENTHALPY) and the maximum number of sequential iterations to be 30.

Listing 2.1: Sequential-Implicit Coupling Keyword

```
COUPLING
SEQ(FLOW{MOLAR_GEO},THERMAL{ENTHALPY},30) /
/
```

The constraints for the flow and thermal equations are specified using the NONLINEAR keyword. Examples of the fixed pressure, fixed density and hybrid constraints are shown in Figure 2.1 2.2 and 2.3 respectively. In all of these examples, we have a constant enthalpy for the flow sequential step. The possible options for the FLOW and THERMAL keywords are: PRESSURE, ENTHALPY, PRES1_DENS_2, DENSITY, DENSITYENTH. These are all interchangeable with each other. Although we can try theoretically 20 different options, only the three specified above are worth describing, as the others do not converge or provide any meaningful results.

```
NONLINEAR
FLOW 1e-4 20 6* ENTHALPY/
THERMAL 1e-4 20 3* PRESSURE/
/
```

Figure 2.1: Fixed pressure constraint keyword specification

```
NONLINEAR
FLOW 1e-4 20 6* ENTHALPY/
THERMAL 1e-4 20 3* DENSITY/
/
```

Figure 2.2: Fixed density constraint keyword specification

```
NONLINEAR
FLOW 1e-4 20 6* ENTHALPY/
THERMAL 1e-4 20 3* PRES1_DENS2/
/
```

Figure 2.3: Hybrid constraint keyword specification

2.4.2 Property Calculation with Constraints

The ADETL library provides the capability to have these different constraints implemented with minimal code duplication. Once the constraint and variable sets have been activated or deactivated using the GENIC framework, the fluid properties are computed for the specified constraints. The geothermal thermodynamic parameters are currently all implemented as a function of the cell pressure and enthalpy ADscalar variables. This is because the original correlations used [15] were formulated with a pressure-enthalpy formulation. To ensure consistent derivative computation for all the fluid properties, the cell pressure and enthalpy variables must first have the correct value and derivatives computed prior to computing the fluid properties. To compute the value, we use a bisection method to compute the pressure or enthalpy value as a function of any two thermodynamic properties. As an example, for the fixed density approach, one of the required steps is to compute the pressure at each inner Newton step for a fixed density (ρ^*). So a bisection method is applied to solve $F(p)$ for p:

$$F(p) = \rho(p, h^k) - \rho^* = 0 \tag{2.16}$$

thus the function to compute the pressure as a function of enthalpy and density is:

$$p(h^k, \rho^*) = \text{BISECTION}(F(p)) \tag{2.17}$$

where:

- $\rho(p, h)$ is the procedure to compute the density as a function of pressure and enthalpy
- h^k is the input enthalpy
- ρ^* is the input density.

Once we have the value, the next step is to ensure that the derivatives are properly populated. This is to ensure that the derivatives of all the other fluid properties are correctly computed, for the constraint that is investigated.

The derivative computation is based on the Inverse Function Theorem, which states that:

$$\frac{dY}{dX} = -\left(\frac{dF}{dY}\right)^{-1} \frac{dF}{dX} \tag{2.18}$$

where

$$F(X, Y) = 0 \tag{2.19}$$

As we require the pressure derivative with respect to enthalpy, we apply the Inverse Function Theorem for $Y := p$, $X := h$, and

$$F(X, Y) = F(p, h) = \rho(p, h) - \rho^* = 0 \tag{2.20}$$

Once the ADscalar pressure and enthalpy for each cell have both their value and derivatives computed, all other fluid properties are computed. The derivatives all will have the correct derivatives since they are all a function of p and h.

2.5 Numerical Comparisons

In the study we considered different problems with varying complexity. The verification of the fully coupled results was shown in [76, 77] where the fully coupled results were compared with analytical, semianalytical and TOUGH2 simulations. Here the focus is to compare the convergence properties of the different sequential formulations with the fully coupled solution. We used a direct linear solver PARDISO [50, 51, 59, 60] to isolate the effects of the linear solver solution on the nonlinear solver. In all the cases, the maximum sequential iterations was set to be 30, the maximum nonlinear iterations was set to 20, the convergence tolerance for the flow and thermal equations was set to 10^{-4}.

2.5.1 One-dimensional Radial Model

This numerical model is a one-dimensional, radial model. The single-phase cases involve the injection or production in the radial model where the fluid remains completely as single-phase liquid water. The two-phase cases involve the propagation of a flash front moving outwards, either a cold-water front for injection or the propagation of a two-phase zone for production.

The rock parameters for this model are defined as:

Table 2.1: Rock and geometry parameters for the one-dimensional radial model

Property	**Value**	**Unit**
Porosity	0.2	%
Permeability	10	md
Thickness	100	m
Rock Compressibility	0	m^3/Pa
Thermal Conductivity	0	W/(m K)

The Corey curves were used as the relative permeability relationships:

$$k_{rw} = (S^*)^4 \tag{2.21}$$

$$k_{rs} = ((1 - S^*)^2(1 - (S^*)^2)) \tag{2.22}$$

$$S^* = \frac{S_w - 0.3}{0.65} \tag{2.23}$$

where k_{rw} is the relative permeability of liquid water and k_{rs} is the relative permeability of the steam.

The same discretization scheme was used as in the code comparison study where the radius of each segment was:

$$r_n = 0.5 \times 2^{n-1}\mathrm{2m}, n = 1, 2, \ldots, 25 \tag{2.24}$$

The pressure and temperature for all the different cases can be seen in Figures 2.4, 2.5, 2.6, 2.7. An example of the AD-GPRS input file can be found in Figure A.1. There is a significant difference in the pressure solutions for the cases that had numerical issues due to different time stepping schemes for each formulation.

Table 2.2: Initial reservoir condition and well constraints for each of the scenarios

Figure Number	2.4	2.5	2.6	2.7
Phases Present	Liquid Water	Liquid Water	Two-phase	Two-phase
Initial Pres (bars)	90	45	90	10
Initial Temp (K)	523.15	523.15	573.15	555.15
Initial Water Sat	1.0	1.0	1.0	0.1
Production rate m^3/s	-	-	14.0	-
BHP well (bar)	45	90	-	90
Injection Temp (K)	-	523.15	-	355.15

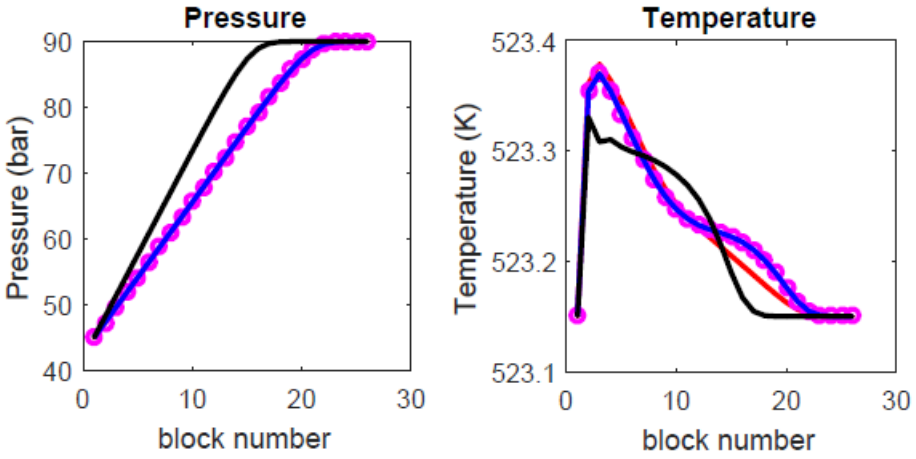


Figure 2.4: Pressure (bar) and temperature (K) for the cells in the single-phase production scenario (black line: fixed density, blue line: fixed pressure, magenta dots: hybrid, red line: fully coupled)

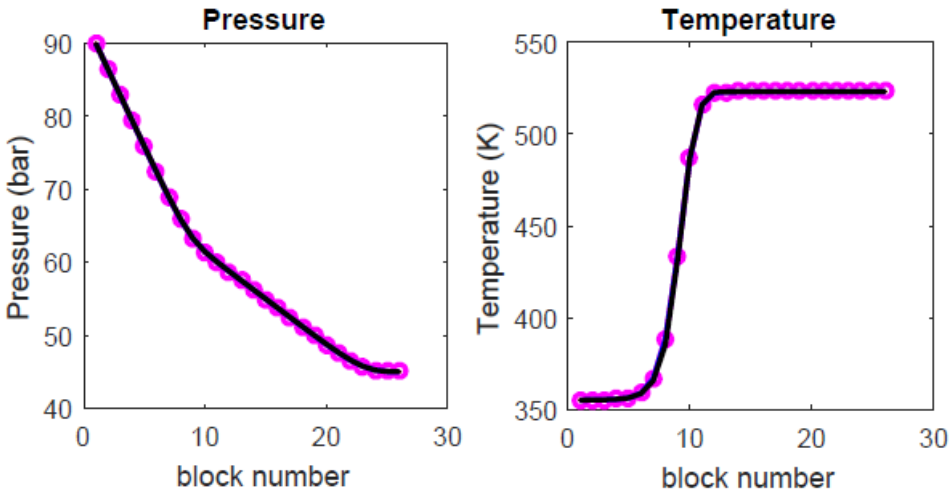


Figure 2.5: Pressure (bar) and temperature (K) for the cells in the single-phase injection scenario (black line: fixed density, blue line: fixed pressure, magenta dots: hybrid, red line: fully coupled)

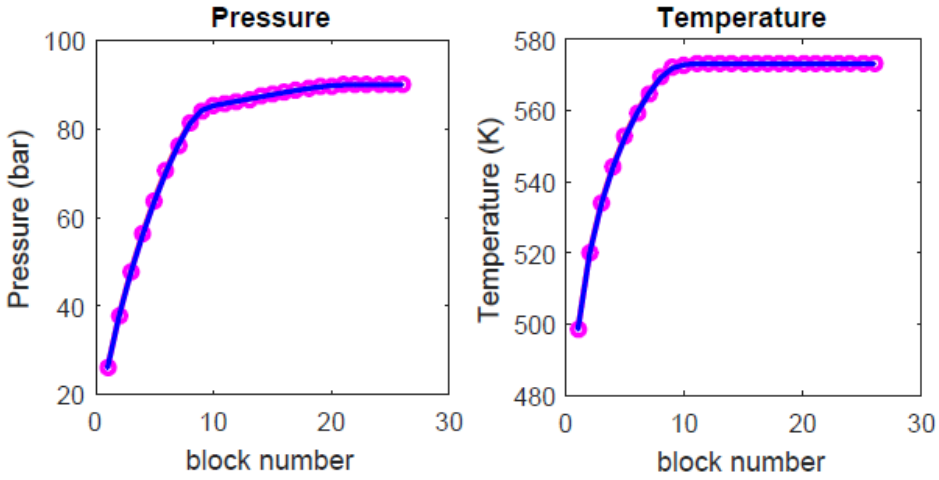


Figure 2.6: Pressure (bar) and temperature (K) for the cells in the two-phase production scenario (black line: fixed density, blue line: fixed pressure, magenta dots: hybrid, red line: fully coupled)

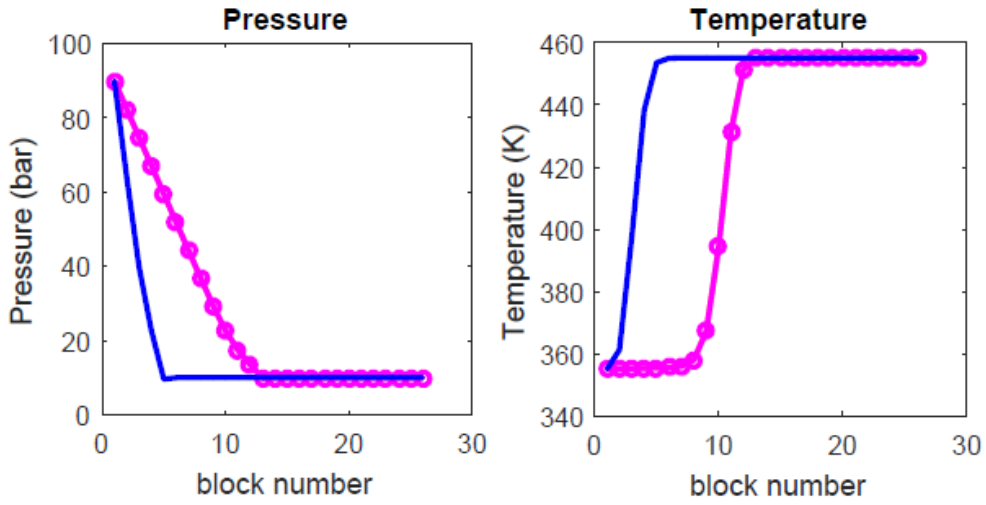


Figure 2.7: Pressure (bar) and temperature (K) for the cells in the two-phase production scenario (black line: fixed density, blue line: fixed pressure, magenta dots: hybrid, red line: fully coupled)

The performance of each of the methods was quantified in terms of the maximum CFL achieved. The CFL number is defined as:

$$CFL = \frac{Q\Delta t}{PV} \tag{2.25}$$

where:

- Q is the throughput in the cell
- Δt is the timestep
- PV is the pore volume

This is a normalized time step with respect to the flow and volume through a cell.

Single-phase Case

In the single-phase production case, the fixed pressure and hybrid scheme achieved identical nonlinear performance. This is because the entire domain remains in single-phase water throughout the entire simulated period, thus a fixed pressure was applied to all the cells. For both the fixed pressure, hybrid and fully coupled method, there were no time step cuts and the predefined maximum CFL of 100 was reached for all schemes. For the fixed density case, the maximum CFL was about an order of magnitude less than the other three schemes.

For the single-phase injection case, the fixed pressure and hybrid scheme obtained identical nonlinear performance. For both the fixed pressure, hybrid and fully coupled methods, there were no time step cuts. For the fixed density case, the maximum CFL was about two orders of magnitude less than the other three schemes.

Table 2.3: Comparison between sequential coupling schemes for single-phase one-dimensional production scenario

	Pressure	**Density**	**Hybrid**	**Fully Coupled**
Number Time Steps	27	62	27	27
Total Newton Iterations	37	2176	37	15
Total Sequential Iterations	50	2285	50	-
Wasted Time Steps	0	40	0	0
Wasted Newton Iterations	0	2637	0	0
MaxCFL	100	12.3	100	100

Table 2.4: Comparison between sequential coupling schemes for single-phase one-dimensional injection scenario

	Pressure	**Density**	**Hybrid**	**Fully Coupled**
Number Time Steps	63	2128	63	63
Total Newton Iterations	694	54228	694	122
Total Sequential Iterations	291	93876	291	-
Wasted Time Steps	0	2119	0	0
Wasted Newton Iterations	0	127015	0	0
MaxCFL	87.7	1.2	87.7	87.7

Two-phase Case

For the two-phase case, the fixed pressure scheme was divergent and the simulation did not converge to a reasonable time step that allowed the simulation to run to completion. The fixed density and fixed hybrid had nearly identical solutions for the production scheme because most of the nonlinearity occurred at the production flash front. For the fully coupled case, the production scenario again did not have any difficulty and there were zero wasted time steps.

For the injection case, the hybrid case performed the best with the largest CFL number and with the lowest number of time step cuts. It is interesting to note that for the injection case, the hybrid method performed better than the fully coupled method. Although the constant density case converges, it is again one order of magnitude worse than the hybrid and fully coupled case. The poor convergence of the fully coupled method will be discussed in further detail in Chapter 4.

Table 2.5: Comparison between sequential coupling schemes for two-phase one-dimensional production scenario

	Pressure	**Density**	**Hybrid**	**Fully Coupled**
Number Time Steps	Divergent	62	61	26
Total Newton Iterations		2176	2078	71
Total Sequential Iterations		2285	2203	-
Wasted Time Steps		40	39	0
Wasted Newton Iterations		2637	2580	0
MaxCFL		9.2	9.2	30.4

Table 2.6: Comparison between sequential coupling schemes for two-phase one-dimensional injection scenario

	Pressure	**Density**	**Hybrid**	**Fully Coupled**
Number Time Steps	Divergent	881	61	79
Total Newton Iterations		23432	906	316
Total Sequential Iterations		38566	373	-
Wasted Time Steps		870	7	31
Wasted Newton Iterations		53104	7	88
MaxCFL		7.4	98.3	93.1

2.5.2 Two-dimensional Heterogeneous Single-phase

The permeability and porosity distributions were taken from the top layer of the SPE10 model, the permeability distribution can be seen in Figure 2 [8]. The grid consisted of $60 \times 60 \times 1$ grid cells, where each grid cell is $20 \times 10 \times 70$m. The scenario was at single-phase throughout the simulation period. Only the single-phase case was tested because the fully coupled case already struggled with the heterogeneous

multiphase solution. The purpose of these test cases was to determine how the hybrid sequential-implicit scheme would compare with the fully coupled case for more complicated scenarios. This is a highly heterogeneous permeability field and tests how the sequential formulation handles strong contrasts in permeability.

Production

For the production scenario, uniform pressure (100 bars), temperature (523.15 K) and water saturation (1.0) were imposed as the initial conditions. A single production bottom hole pressure (45 bars) controlled well was located in the center of the reservoir. The production simulation was run for 100 days.

Injection

For the injection scenario, uniform pressure (45 bars), temperature (523.15 K) and water saturation (1.0) were imposed as the initial conditions. A single injection bottom hole pressure (100 bars) controlled well was located in the center of the reservoir. The injection simulation was run for 500 days. An example of the AD-GPRS input file can be found in Figure A.2.

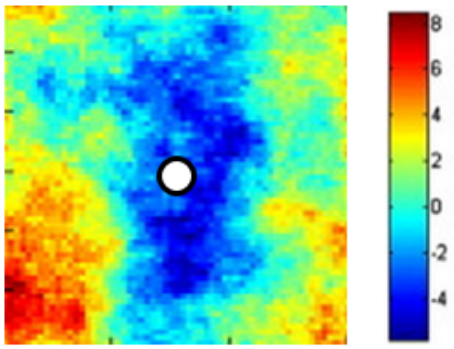


Figure 2.8: Log of permeability (md) distribution of a section of the top layer of the SPE 10 model (White dot represents the production or injection well)

Results

The hybrid method had worse results in the two-dimensional heterogeneous case than the one-dimensional homogeneous test cases. For the production scenario, the results were comparable (MaxCFL of 30.1 compared with 46.3) to the fully coupled case. However, for the injection scenario, the fully coupled case was two orders of magnitude better than the hybrid method. It is clear from these two cases that the sequential formulation performed worse than the fully coupled case for this two-dimensional heterogeneous case.

Table 2.7: Comparison between the sequential and fully coupled method for single-phase two-dimensional heterogeneous production scenario

	Hybrid	**Fully Coupled**
Number Time Steps	42	30
Total Newton Iterations	76	29
Total Sequential Iterations	487	-
Wasted Time Steps	13	0
Wasted Newton Iterations	624	0
MaxCFL	30.12	46.31

Table 2.8: Comparison between the sequential and fully coupled method for single-phase two-dimensional heterogeneous injection scenario

	Hybrid	**Fully Coupled**
Number Time Steps	774	51
Total Newton Iterations	15421	68
Total Sequential Iterations	31814	-
Wasted Time Steps	741	0
Wasted Newton Iterations	30938	0
MaxCFL	1.43	100.4

2.5.3 Three-dimensional model with phase change

The final case was a three-dimensional model based on a reservoir model from the 1980 Code Comparison Study [62]. The reservoir model consists of single-phase liquid water with a two-phase zone of immobile steam sandwiched between a hot and cold water region. Production is performed from a well that is completed below the two-phase zone. The parameters were chosen such that the boiling in the well occurs after

a certain period of production. Although the parameters are relatively homogeneous, this model was defined to be a prototype for field-wide reservoir development studies. The rock properties for this model can be found in Table 2.9. This problem has both three-dimensional flow with phase transitions and two-phase flow, including gravity drainage. A full discussion on the problem description can be found in [62]. The computational grid was set to $20 \times 25 \times 30$ total cells that are $200 \times 200 \times 60$ m in size. The production well was completed in the corner cell and perforates layers 16-20. The water rate at 100m^3/day was used as the well control. The simulation time was for 500 days. An example of the AD-GPRS input file can be found in Figure A.3.

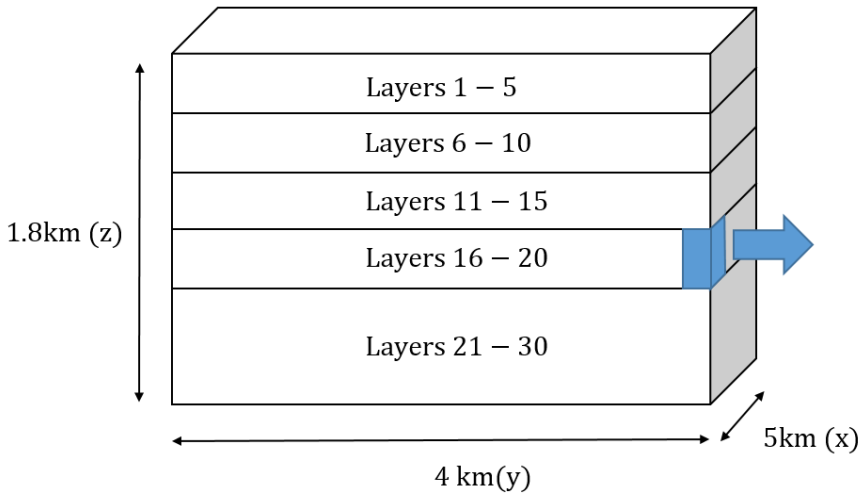


Figure 2.9: Schematic of the three-dimensional model

Table 2.9: Reservoir properties for the 3D Two-phase Gravity Drainage Problem

Parameter	**Value**	**Unit**
Reservoir dimensions	$4000 \times 5000 \times 1800$	m
Porosity in each layer	[0.2,0.25,0.25,0.25,0.2,0.2]	-
Layer permeability in x and y directions	[100,200,200,200,100,100]	md
Layer permeability in z direction	[2,50,50,50,2,2]	md
Layer gas saturation	[0,0.15,0,0,0,0]	-
Depth at layer top	[150,450,750,1050,1350,1650]	m
Temperature at layer top	[433.15,553.15,. . . ,553.15]	K
Pressure at layer top	[40,64,88,112,136,160]	bar

Similar to the two-dimensional heterogeneous case, the hybrid formulation performs worse than the fully coupled case (Table 2.10). Again the performance is about two orders of magnitude worse than the fully coupled case. This can be observed by noting the maximum CFL of about two orders of magnitude worse than the fully coupled case. It is important to note that the fully coupled case did not have any wasted time steps and a more aggressive time stepping scheme could actually provide better results for the fully coupled method.

Table 2.10: Comparison between the sequential and fully coupled method for three-dimensional model

	Hybrid	**Fully Coupled**
Number Time Steps	1392	21
Total Newton Iterations	1217	35
Total Sequential Iterations	4569	0
Wasted Time Steps	1382	0
Wasted Newton Iterations	29944	0
MaxCFL	0.4	11.9

2.6 Sequential-implicit Summary

Using a general implicit coupling framework, we investigated the convergence properties for three different sequential coupling schemes. It was found that a hybrid method where a fixed pressure for single-phase cells and fixed density for two-phase cells performed the best out of the three that were tested. This hybrid method was able to avoid the divergent or slow converging behavior in the single-phase (fixed density) or two-phase (fixed pressure) regions. More numerical tests also showed that the fully coupled method performed significantly better than the sequential scheme. This difference was more pronounced for the two-dimensional and three-dimensional models. This poor performance is attributed to the slow outer loop convergence of the sequential method. This shows that for these more complex problems, the coupling between the flow and thermal equations is too strong and thus the standard sequential-implicit scheme is insufficient to model the strongly coupled nature of the problem.

2.7 Modified Sequential-Fully Implicit Method

From the earlier sections of this chapter, we see that the sequential-implicit method for geothermal simulations has very poor convergence for complex two-dimensional and three-dimensional scenarios. This is because of the strong coupling between the flow and thermal equations thus leading to slow outer loop convergence rate. A similar difficulty was encountered in flow and transport problems. To address this issue, [42] developed a modified-sequential fully implicit (m-SFI) method that enriched the pressure equation for all the strongly coupled cells. This enrichment involved solving additional transport equations during the pressure solve for the strongly coupled cells. The following sections in this chapter describe the investigations of an m-SFI method for geothermal simulation. We also present numerical comparisons of different choices of the subdomain for this m-SFI method.

2.7.1 Algorithm

For the modified sequential fully implicit formulation (m-SFI), the flow and thermal residual equations are solved separately for their respective primary variables x_F, x_T. The main difference between the modified sequential fully implicit method and the sequential-implicit method is that in the first step where the flow residual is solved, a subset of the thermal residual equations is included in this step. For this study, we only examined a sequential formulation for a pressure-enthalpy formulation as described in [77]. The pressure-enthalpy formulation was selected as it does not require any variable switching and thus allows for a simpler implementation and analysis for this method. It is important to note that this sequential formulation is iteratively coupled until convergence is reached for both the mass and energy conservation equations. The steps for the m-SFI are as follows:

Step 1: Solve: For cells in M_1:

$$R_F^{n+1}(x_F^{n+1}, x_T^{n+1}) = 0 \tag{2.26}$$

$$R_T^{n+1}(x_F^{n+1}, x_T^{n+1}) = 0 \tag{2.27}$$

For the remaining cells not in M_1:

$$R_F^{n+1}(x_F^{n+1}, x_T^{n+1}(\partial h^{n+1} = 0)) = 0 \tag{2.28}$$

This involves solving the flow residual and constraining the enthalpy to be fixed ($\delta h = 0$) for the cells that are not in the domain M_1. Because all of these cells are at single-phase, a fixed enthalpy was chosen. Once we have the solution $x_F^{n+1,k^*}, x_T^{n+1,k^*}$ we use that as the initial guess to the thermal equation.

Step 2: Solve for the cells in M_2:

$$R_T^{n+1}(x_F(\partial p = 0), x_T^{n+1}) = 0 \tag{2.29}$$

This involves solving the thermal residual equation for the cells in M_2 while constraining the pressure to be fixed in those cells ($\partial p = 0$). The choice of M_1 and M_2 was a focus in this work. It is important to note that at both steps we are solving a system of nonlinear equations using Newton's method. These two steps, Step 1 (Equation 2.26 and 2.27) and Step 2 (Equation 2.29) are repeated sequentially until convergence for both residual equations are reached simultaneously.

Subdomain Definition

We define M_1 as the subdomain that has a strong coupling between the flow and thermal equations and M_2 as the weakly coupled domain. In this work, we investigated different definitions of the strongly coupled and weakly coupled domains. We measured the coupling strength based on the Courant-Friedrichs-Lewy (CFL) of the cell and the phase state of the cell:

$$\text{CFL} = \frac{\Delta t Q}{PV} \tag{2.30}$$

where:

- Δt is the time step
- Q is the volume injection rate
- PV is the pore volume of the cell

We define a cell i to be in M_1 if any of the following conditions are satisfied:

- $\text{CFL}_i > \text{CFL}_{\text{tol}}$ OR Number of phases in cell i is greater than 1 OR A well perforation penetrates cell i

We also investigated adding all the neighboring cells that satisfy this criterion. This was to limit the discontinuities between the flow and thermal residual equations. We refer to this strategy as m-SFI-N.

The criterion for M_2 is the complement of M_1:

- $\text{CFL}_i < \text{CFL}_{\text{tol}}$ AND Number Phases in cell i is 1 AND no well perforation penetrates cell i

2.8 AD-GPRS

2.8.1 Sequential Framework

The key difference in the sequential-implicit and m-SFI algorithms is that for the m-SFI approach, we include some of the thermal residual equations when solving for the flow residual equations. To specify this type of input we utilize mappers that are available in the GENIC. The mappers allow for a general specification of which cells to include at each solution step. Every subproblem has a default mapper of including all cells. We specify these mappers with the COUPLING keyword. An example of this is shown in Figure 2.10.

```
COUPLING
SEQ(FIM(FLOW{MOLAR_GEO},THERMAL{ENTHALPY}<MAP_MSFI_T1>),
THERMAL{ENTHALPY}<MAP_MSFI_T2>,30) /
/
```

Figure 2.10: m-SFI Coupling Keyword

Here we specify that at the first subloop, we solve for all the cells for the flow equation and include all the thermal residual equations that correspond to cells in the MAP_MSFI_T1 mapper. In the second subloop, we solve for all the thermal residual equations that correspond to the cells defined by the MAP_MSFI_T2 mapper.

2.8.2 Mappers

The main function of each mapper is to specify which cells are active. This specification is updated before the Jacobian matrix is formed. An example of this update is shown in Figure 2.11. Here we see the three checks that are done, first whether it is two-phase, the CFL exceeds the CFL tolerance or if it is in a well cell. If any of these

conditions are met, we append the cell index into the vector of indices m_actives. This contains the active cells of the variable set that this mapper is applied to. So for the example shown in Figure 2.10, this would specify that all the cells that have: two or more phases, a CFL higher than the CFL tolerance or is a well cell are also active.

```
virtual void updateMapping(){
    double cfl_max = thermalNonlinearParams.cfl_max;
    m_actives.clear();
    m_actives.reserve(therm_cell_subset->num_blocks());

    for (size_t ib = 0; ib < therm_cell_subset->num_blocks(); ++ib){
      // If it is two-phase or greater than cfl than then it is active
      if (flow_cell_subset.num_phases >= 2
        || flow_cell_subset->cfl(ib) > cfl_max)
        m_actives.push_back(ib);

      else if (!SimData::reference().wells_markers.empty()){
        If it is a well block
        if (SimData::reference().wells_markers[ib] != 0)
          m_actives.push_back(ib);
      }
    }
    update_is_active();
};
```

Figure 2.11: Code example of mapper update for m-SFI

This type of mapper specification is general and allows the user of the GENIC

framework to design mappers for a wide range of applications. For example, this could also be used for nonlinear domain decomposition techniques, where each section in the domain is solved separately and sequentially coupled.

2.9 Numerical Comparisons

First, we considered the same numerical model described earlier (Section 2.5.3) to see whether this m-SFI method improves the convergence of sequential-implicit scheme. To decouple the effects of the linear solver solution on the nonlinear solver, a direct solver PARDISO [50, 51, 59] was used for the linear solver. In all the cases, the maximum sequential iterations was set to be 30, the maximum nonlinear iterations was set to 20, and the convergence tolerance for the flow and thermal equations was set to 10^{-4}. However, the use of a direct solver limits any strong conclusions on the computational speed-up.

2.9.1 Three-dimensional model with phase change

Model Description and Verification

The first test case was a three-dimensional model based on a reservoir model from the 1980 Code Comparison Study [62]. This is the same reservoir model as described in Section 2.5.3. The key difference, in this case, is the refinement of the grid and the simulation time that was changed. This was to stress the nonlinear solver of the methods to allow for an accurate representation of the full capabilities of all the formulations. An example of the AD-GPRS input file can be found in Figure A.3.

The final pressure and temperature distributions are shown in Figure 2.12. The relative difference between the pressure and temperature for fully coupled method and m-SFI is shown in Figure 2.13. We can see that there is a very close agreement

where the difference is at most 0.01%. This is expected because the fully coupled and m-SFI method both require the flow and thermal residual equations to be below a convergence tolerance, thus if converged they should have a close agreement.

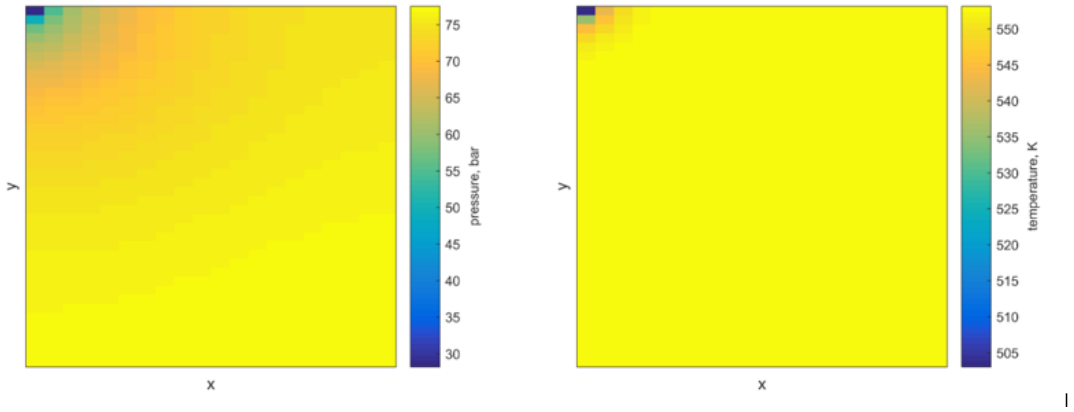


Figure 2.12: 3D Pressure and temperature distribution after 10 years of production at 16th layer for $20 \times 40 \times 36$ cells

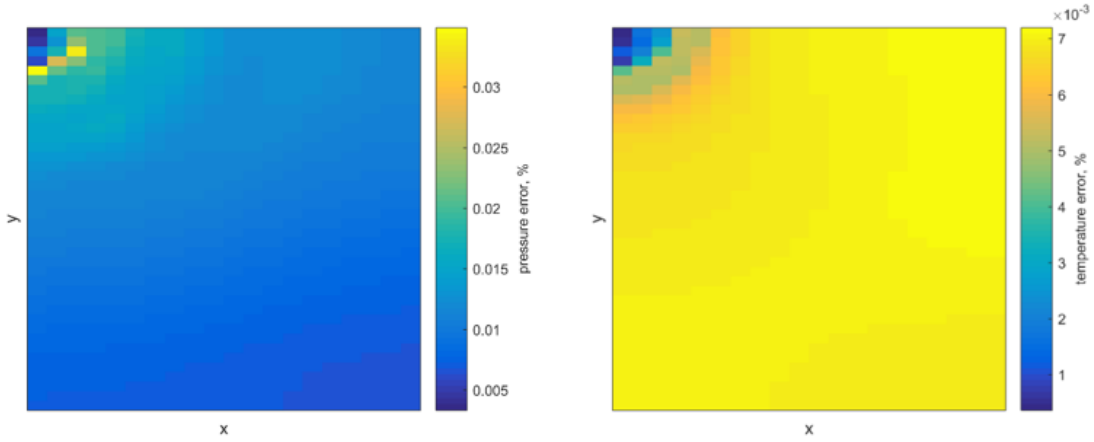


Figure 2.13: Relative error between the fully coupled and m-SFI-N approach for pressure and temperature after 10 years at 16th layer for $20 \times 40 \times 36$ cells

Nonlinear Performance

The focus of this section is to compare the performance of the m-SFI method with the fully coupled method. Table 2.11 shows the nonlinear solver performance for the different strategies tested. We see that for this test case, regardless of the CFL_{tol} selected, the m-SFI methods converge to the same maximum CFL (maximum CFL is defined as the maximum CFL for all the cells for all the time steps) as the fully coupled method and has less wasted time steps than the fully coupled method. We also see that the m-SFI method is able to overcome the limitations that the sequential-implicit (SI) had for complex problems and decreased the number of sequential iterations per timestep by 10 times. For this specific case, a CFL_{tol} of [0.1,1,10] all resulted in about the same performance. This is because, at this grid resolution, the complexity is not sufficient to demonstrate the effects of different CFL_{tol} values or the inclusion of neighboring cells.

Table 2.11: Comparison of the nonlinear solver performance for the different coupling schemes for $10 \times 20 \times 18$ cells (FC: Fully Coupled, SI: Sequential-implicit)

	FC	**SI**	**m-SFI**			**m-SFI-N**
CFL_{tol}	**-**	**-**	**0.1**	**1**	**10**	**1**
Timesteps	40	215	39	39	39	39
Total Newton Iterations	136	5050	306	308	324	213
Total sequential iterations	0	6681	134	136	137	105
Seq iterations/timesteps	-	31.1	3.4	3.5	3.5	2.7
Number timesteps wasted	4	201	1	1	1	1
% of cells in M_1	100	0	27.89	19.25	19.06	30.30
Maximum CFL	5.343	2.707	5.342	5.342	5.342	5.342

The original grid was coarsened and refined to test how the m-SFI method performs for different CFL_{tol}. Here we looked at a low resolution ($4 \times 5 \times 6$ cells) and a high resolution ($20 \times 40 \times 36$ cells) grid. Table 2.12 shows the results of the different resolutions for the different methods. For the low resolution, we see that similar to the previous case, the m-SFI methods all have the same time stepping result and converge for the same maximum CFL. However, for the refined case (H), only the m-SFI-N method had the same time stepping result as the fully coupled method. We notice that regardless of the CFL_{tol} selected, the number of time steps is still higher than the fully coupled and m-SFI-N method. m-SFI-N had a greater impact on the nonlinear performance for the refined case as now there is a sharper boundary between the single- and two-phase regions. Thus, having the additional layer of cells would improve that coupling at that boundary. We note that for the high-resolution grid, the computational time of the m-SFI-N method was 10% faster than the fully coupled method. This computational speedup is a combination of the relatively low number of sequential iterations per time step and small % of cells in M_1. As mentioned earlier, since only a direct linear solver is used, it is difficult to conclude that the m-SFI-N can indeed outperform the fully coupled method.

Table 2.12: Comparison of the nonlinear solver performance for two-phase gravity drainage problem at different grid resolutions (L: Low, $4 \times 5 \times 6$ cells, H: High, $20 \times 40 \times 36$)

	FC		**m-SFI**						**m-SFI-N**	
CFL_{tol}	**-**		**0.1**		**1.0**		**1.0**		**1.0**	
Grid resolution	**L**	**H**	**L**	**H**	**L**	**H**	**L**	**H**	**L**	**H**
Timesteps	19	162	19	584	19	592	19	592	19	162
Newton Iter.	28	454	67	1982	67	2004	67	2016	41	585
Sequential Iter.	-	-	49	1901	49	1918	49	1946	40	470
Seq iter./timesteps	2.6	3.3	2.6	3.2	2.6	3.3	2.1	2.9	2.6	3.3
Timesteps wasted	0	1	0	433	0	436	0	436	0	1
% of cells in M_1	100	100	19.9	16.8	17.5	15.8	17.5	15.2	50.8	24.4
Max. CFL	0.51	4.07	0.51	4.0	0.51	4.07	0.5	4.07	0.51	4.07

2.9.2 SPE 10 Three-dimensional Problem

The next problem we considered was a three-dimensional problem with strong geological heterogeneity. This is an extension of the two-dimensional problem in Section 2.5.2, but with a larger domain and three-dimensional flow. Here the focus will be to compare the convergence properties of the m-SFI method with the fully coupled solution. The permeability and porosity distribution (Figure 5) were taken from the top four layers of the SPE 10 test case problem [8], Table 2.14 shows the reservoir properties for this SPE10 problem. For this test case, the domain was discretized using $60 \times 220 \times 4$ cells. Uniform pressure (50 bar), temperature (523.26 K) and water saturation (1.0) was imposed as the initial conditions. There were four wells specified in this problem, two injectors ($p = 90$ bar, $T = 433.15$K) and two producers ($p = 30$ bar) all operating under bottom hole pressure control. This problem is highly

heterogeneous and follows a Gaussian distribution with high contrast and channelized rock structure. Due to this heterogeneity, this is challenging even for the fully coupled methods and has convergence issues for more aggressive time step schemes. The final pressure and temperature solution for the simulated period is presented in Figure 2.16.

Figure 2.14: Reservoir properties for 3D SPE 10 Problem

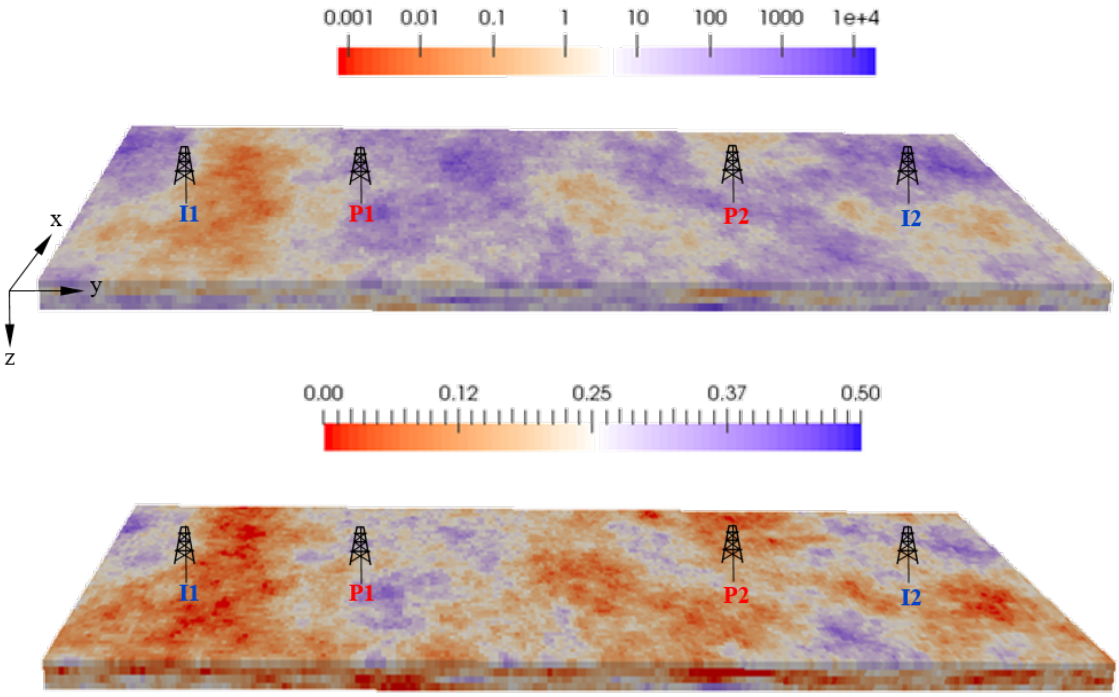


Figure 2.15: Schematic of the three-dimensional model

Parameter	Value	Unit
Reservoir dimensions	$365.8 \times 670.6 \times 51.8$	m
Initial reservoir pressure	50	bar
Initial reservoir	523.26	K
BHP	30 for producers 90 for injectors	bar
Injection temperature	433.15	K
Relative phase permeability	$k_r j = S_j^2$	-
Rock thermal expansion	2×10^{-5}	1/K
Rock heat capacity	2000.0	kJ/(kg K)
Rock thermal conductivity	150.0	kJ/(m day K)
Water thermal conductivity	53.5	kJ/ (m day K)
Gas thermal conductivity	3.59	kJ/(kg K)

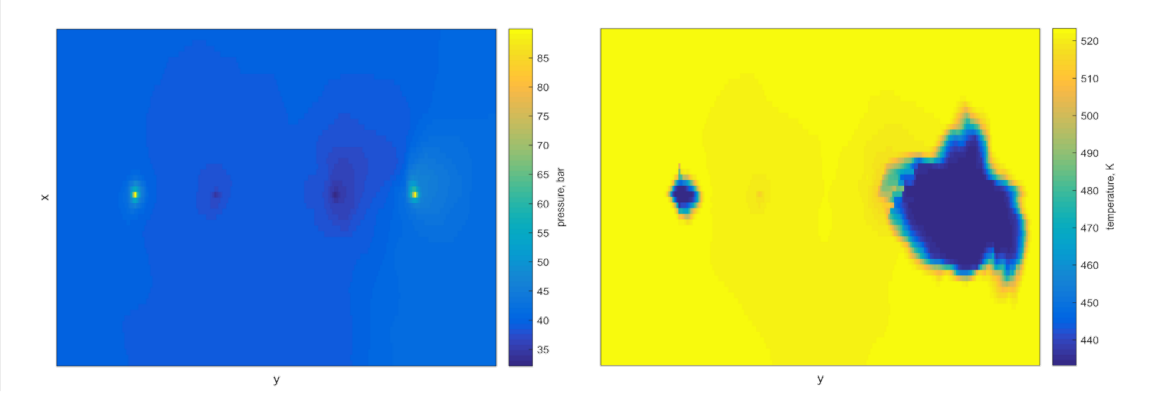


Figure 2.16: Pressure (left) and temperature (right) distribution for $60 \times 220 \times 4$ grid after 100 days at top layer

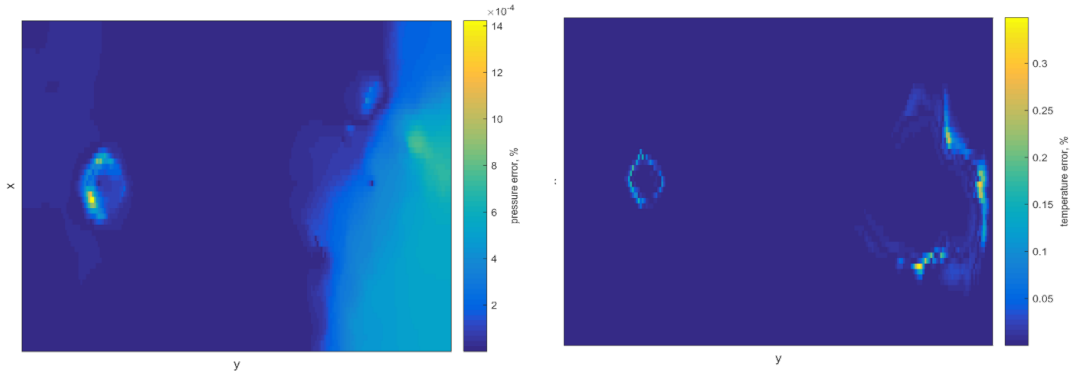


Figure 2.17: Relative difference between the fully coupled and m-SFI-N approach for pressure (left) and temperature (right) distribution for $60 \times 220 \times 4$ grid after 100 days at top layer

We see in Figure 2.17, the relative difference between the fully coupled and m-SFI-N method is slightly higher than the previous two-phase gravity drainage problem. For this specific scenario, the error is $10^{-3}\%$ for pressure and at most 0.5% for the temperature. This error is localized at the temperature front.

Nonlinear Convergence Comparison

Table 2.13: Comparison of the nonlinear solver performance for the different coupling schemes for $30 \times 110 \times 2$ cells (FC: Fully Coupled)

	FC	**m-SFI**	**m-SFI**	**m-SFI-N**
$\mathbf{CFL_{tol}}$	-	0.1	1.0	0.1
Number of timesteps	53	75	111	53
Total Newton iterations	179	1869	3974	793
Total sequential iterations	-	797	1599	377
Sequential iterations/timesteps	-	10.6	14.4	7.1
Number of timesteps wasted	9	37	81	8‘
% of cells in M_1	100	71.75	74.81	67.42
Normalized Run time	1	8.3	17.43	3.15
Maximum CFL	44.13	44.24	42.04	44.18

Table 2.13 shows the results of the different methods for a grid structure of $30 \times 110 \times 2$ cells. Here we notice that the m-SFI-N at a $\mathrm{CFL_{tol}}$ of 0.1 requires about half the number of time steps in comparison to a $\mathrm{CFL_{tol}}$ of 1. This is because, with a smaller CFL, this means we have a larger number of cells in each iteration thus are able to capture the front better. We see that the percentage of cells is larger for the case where $\mathrm{CFL_{tol}} = 1$; this is because of the number of time step cuts and thus more iterations are spent where the front has spread. Due to the injection and production fronts in this problem, we notice that m-SFI ($\mathrm{CFL_{tol}}$=0.1) performs better than m-SFI-N ($\mathrm{CFL_{tol}}$=1) due to the need to predict where the front is.

Table 2.14 shows the results for a low ($15 \times 55 \times 1$) and high ($60 \times 220 \times 4$) grid resolution. In order to coarsen and refine the grid, volume averaged properties were used for the porosity and permeability. Here we see that for a low resolution, all four

of the methods result in the same number of timesteps to complete the simulation. For the high grid resolution, only the m-SFI-N method had a comparable maximum CFL and a similar number of time steps. However, all the m-SFI cases are four times slower compared to the fully coupled method. This is due to the higher number of sequential iterations per time step than in the previous test case and the larger percentage of cells in the M_1 subdomain, thus having a larger cost per sequential step. In summary, we have found that for this case where there is a large proportion of two-phase cells, the m-SFI method performs poorly, because most cells have a strong coupling, there is not much gain in splitting the flow and thermal equations sequentially.

Table 2.14: Comparison of the nonlinear solver performance for 3D SPE 10 Problem at different grid resolutions (L: Low, H: High)

	Fully Coupled		**m-SFI**		**m-SFI**		**m-SFI-N**	
	L	**H**	**L**	**H**	**L**	**H**	**L**	**H**
CFL_{tol}	-		0.1		1.0		0.1	
Number of timesteps	49	114	49	126	49	177	49	115
Newton iterations	77	443	708	4077	1061	8203	708	3412
Sequential Iterations	-	-	371	1509	476	2969	371	1323
Seq iterations/timesteps	-	-	7.6	12.0	9.7	16.8	7.6	11.5
Timesteps wasted	0	19	0	40	0	106	0	21
% of cells in M_1	100	100	55.18	70.62	22.13	68.6	55.2	74.2
Normalized run time	1	1	4.92	4.83	6.69	10.2	4.91	4.15
Maximum CFL	59.4	62.4	59.4	62.2	59.4	62.7	59.4	62.4

2.10 Chapter Summary

In work described in this chapter, we investigated two key aspects of sequential-implicit simulations. First, we identified a sequential-implicit scheme that was convergent for both single- and two-phase simulations. It was found that a hybrid approach was necessary, where a fixed pressure was applied to single-phase cells and a fixed density for two-phase cells. However, from numerical comparisons, it was shown that the numerical convergence for the hybrid method on more complex scenarios still performed poorly in comparison to the fully coupled method.

To improve on the sequential outer loop convergence, a modified sequential fully implicit (m-SFI) method was applied. This was based on the work by [42] for flow and transport problems. Here we found that the coupling strength of the problem is determined by both the phase-state and the CFL of the cell. We were able to achieve comparable convergence profile when the $\mathrm{CFL}_{\mathrm{tol}}$ is small (0.1). However, for these scenarios, we found that since a large number of cells would need to be included, the computational gains from solving them separately is diminished, thus the computational run time is often slower than the fully coupled method.

From this work, we conclude that the current research available on sequential schemes for geothermal applications is insufficient. Due to the strong coupling between the flow and thermal equations, the sequential-implicit method has poor performance compared to the fully coupled problem. In order to make sequential-implicit schemes a viable option, the outer loop convergence of these schemes must be improved.

Chapter 3

Sequential-implicit Newton's Method

3.1 Introduction

Efficient simulation of multiphysics problems such as geothermal simulation is a challenging task. This is often due to the different complexities and coupling inherent in these problems. One approach to this problem is to solve the entire multiphysics problem simultaneously in a fully coupled manner. However, due to the strong coupling and multiphysics interactions it is difficult to analyze and challenging to design solvers for these fully coupled methods. Instead, the sequential-implicit method splits the multiphysics problem into different subproblems that are each solved separately. The isolation of each subproblem allows for specialized solvers and simulators to be designed to tackle each complexity separately. Once each subproblem is solved in an efficient manner, the entire multiphysics problem is coupled through a sequential-implicit scheme. However, as shown in the previous chapter, sequential-implicit methods can suffer from slow convergence when there is a strong coupling between the individual subproblems. This slow convergence is a result of the linear convergence rate of the

fixed-point iteration that is used in sequential-implicit methods.

This chapter addresses the further improvement of the efficiency of sequential-implicit fixed point (SIFP) methods for geothermal simulation and coupled flow-mechanics problems. SIFP methods have proven to be attractive for solving flow-mechanics problems, where a fully coupled approach have shown to have significant linear solver scalability issues [74, 75, 7, 31]. A sequential strategy reduces the complexities and requirements of the linear solver, thus making them easily scalable in comparison to a fully coupled approach. A sequential coupling in flow and mechanics can also be beneficial when they have different computational domain sizes (geomechanics domain is often much larger than flow domain), different spatial discretization schemes (finite volume for flow and finite element for mechanics) or different simulators for flow and mechanics (e.g. TOUGH2 [54] for flow and ABAQUS for mechanics [56]). To improve these sequential methods, the coupling between the flow and mechanics problem has been investigated extensively. It was shown that the fixed-stress and undrained split are unconditionally stable and that the fixed stress converges faster than undrained split [28, 30, 29, 40]. However, the performance of the fixed stress method is strongly dependent on the coupling strength between the flow and mechanics problems [28]. If the coupling strength is too high, this could result in a slow convergence rate thus requiring too many sequential iterations to couple the two problems together. This would limit the capability of these sequential-implicit methods.

When modeling isothermal multiphase flow and transport in porous media, a sequential strategy has also proven to be useful. Jenny et al. [26] first used this sequential strategy in a Multiscale Finite Volume formulation to simulate immiscible multiphase flow in porous media. The sequential solution strategy was essential to the multiscale formulation where the near-elliptic flow and hyperbolic transport problems are solved separately. This separation of the elliptic and hyperbolic parts allowed for

these advanced discretization schemes and scalable algorithms to be utilized. However, similar to the flow and mechanics problem, due to their strong coupling, this scheme can also require a large number of outer iterations. This would overshadow the computational gains from using these specialized solvers. There has been significant work in trying to improve this separation to decrease the number of outer iterations [42, 33, 44, 45, 44, 43]. Their approaches involved modifying the equations of each subproblem to reduce the splitting errors, as a result, reducing the number of sequential iteration. It was found that the specific details of how each physical problem is linearized, split and formulated have strong contributions to the splitting-error of the sequential scheme [43] and thus the number of sequential iterations. To reduce this splitting error, [42] used the phase appearance and disappearance of the grid cells to determine how to enrich the pressure equation.

In the work described in the previous chapter, we investigated the sequential-implicit method for the flow and thermal problem in geothermal reservoir simulation. Inspired by the different splitting strategies in flow and mechanics, we investigated applying different constraints when solving the flow and thermal equations. We found that a naive splitting of a fixed enthalpy when solving the flow equation and fixed pressure when solving thermal equation converged for single-phase cells but diverged for two-phase cells. As a result, we developed a hybrid method where a fixed pressure was used for single-phase cells and fixed density for two-phase cells. Although this proved to be the best out of the sequential schemes examined, it still suffered from a large number of outer loop iterations for strongly coupled flow and thermal problems. Similar to [42], to improve the outer loop convergence, we enriched the flow equations based on the phase state of the cells. This improved the convergence of the outer loop for the sequential formulation, however, these additional equations increased the cost of each sequential iteration that could negate the benefits of solving it in a sequential manner.

We observed that the main bottleneck in SIFP methods lies in the outer fixed-point loop, which typically converges linearly (i.e., the error decreases to zero like a geometric sequence). To obtain superlinear convergence, one way is to apply Anderson acceleration [1], which is essentially a nonlinear analog of GMRES [57]; it is also related to multisecant quasi-Newton methods [70] and is convergent when the underlying fixed point map is a contraction [64]. Such an acceleration has been applied successfully for fixed-stress splitting schemes for nonlinear poromechanics of unsaturated materials [5]. Another approach is to use the fixed point map as a preconditioner to Newton's method, which is the idea behind the multiplicative Schwarz preconditioned inexact Newton (MSPIN) algorithm proposed in [37]. MSPIN uses a partitioning of the primary variables by field type, and solve for the groups of variables successively in a nonlinear multiplicative Schwarz manner. This mapping is then used to precondition Newton's method with an approximate Jacobian. MSPIN has been shown to be effective for high Reynolds number Navier-Stokes problems; however, as already observed in [37], "the determination of the partition of the physical variables can be the most interesting part of implementation, because the best choice is generally problem-specific." Our own observations indicate that, in addition to the choice of variable sets, the coupling conditions between these sets are equally important to the efficiency of method, just like for SIFP methods.

Thus, our contribution in this work is to improve on the MSPIN approach by identifying an effective variable splitting and coupling conditions for the flow-thermal and flow-mechanics problems. Our new method, called the Sequential Implicit Newton (SIN) method, is constructed by using SIFP as a preconditioner to Newton's method. One key difference of this method from MSPIN is the coupling conditions, which we implement by augmenting the sequential subproblems with physically motivated constraints. A second key difference is that we use an exact Jacobian in computing our Newton update, rather than an inexact Jacobian as in MSPIN; a similar approach

was used for the restricted additive Schwarz preconditioned exact Newton method (RASPEN) [13], but here we implemented it in a multiplicative Schwarz fashion. We showed that matrix-vector multiplication involving the exact Jacobian can be performed by reusing matrix factorizations done at earlier steps of the algorithm. Thus, our method enjoys both the local quadratic convergence of an exact Newton, but at a relatively low computational cost. This leads to a significant improvement over the unaccelerated SIFP methods, which we confirm by our numerical experiments.

The rest of the chapter is organized as follows. In Section 3.2, we describe the sequential-implicit Newton method in detail for an abstract nonlinear system. In particular, we explain how the Jacobian matrix-vector product, which is required for calculating the Newton update using GMRES, can be calculated using matrix factorizations already computed when solving the sequential subproblems. In Section 3.4, we present the geothermal flow-thermal problem and show how to implement the SIN method with the specific variable partition and coupling conditions suggested by the previous chapter for the SIFP method. We also show numerical results illustrating the improvements over the SIFP method. We do the same in Section 3.5, but for the flow-mechanics problem. Chapter conclusions are given in Section 3.6.

3.2 Sequential-implicit Newton's Method

In this section, we show the derivation of the sequential-implicit Newton's method using an existing sequential iteration for a general multiphysics problem. For simplicity and clarity of exposition, assume that the multiphysics problem is given by a set of residual equations $R(x) = 0$ that can be split into $R(x) = (R_1(x), R_2(x))$, where R_1 and R_2 are nonlinear functions that represent different physical processes. Moreover, the unknowns themselves can be split into $x = (x_1, x_2)$, with the two groups of unknowns potentially following different dynamics. Then the fully coupled (FC)

method is simply Newton's method applied to the entire multiphysics problem, with each Newton update obtained by solving the linear system:

$$\begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \delta x_1^k \\ \delta x_2^k \end{bmatrix} = - \begin{bmatrix} R_1(x^k) \\ R_2(x^k) \end{bmatrix}, \tag{3.1}$$

for $(\delta x_1^k, \delta x_2^k)$, where $J_{ij} = \frac{\partial R_i}{\partial x_j}$ are the are the partial derivatives of the i^{th} subproblem residual equation with respect to the j^{th} subproblem primary variable set. The next iterate x^{k+1} is then defined by:

$$x_i^{k+1} = x_i^k + \delta x_i^k, \qquad i = 1, 2.$$

Although the focus of this study was to compare the sequential-implicit Newton method with the fixed-point iteration, it is useful to compare with this method to understand the nonlinear behavior of the underlying coupled nonlinear problem.

3.2.1 Sequential-implicit Fixed Point Method

The sequential-implicit fixed point (SIFP) method involves partitioning the overall problem into multiple subproblems that are solved sequentially. In the case of a multiphysics problem, each of these subproblems will often correspond to a specific physical problem. Here we consider a splitting strategy that involves two different subproblems and an auxiliary constraint applied to each of the solution steps. The sequential-implicit scheme begins with an initial guess $x^0 = (x_1^0, x_2^0)$ for the entire problem. Here x_1 and x_2 correspond to the variable set for the two different subproblems. To solve for the next time step, the sequential process below is followed:

1. Solve:
$$\begin{cases} R_1(x_1^*, x_2^*) = 0, \\ \quad b(x_1^*, x_2^*) = b(x_1^k, x_2^k), \end{cases} \tag{3.2}$$
for (x_1^*, x_2^*), using e.g. Newton's method.

2. Solve:
$$\begin{cases} R_2(x_1^{**}, x_2^{**}) = 0, \\ \quad c(x_1^{**}, x_2^{**}) = c(x_1^*, x_2^*), \end{cases} \tag{3.3}$$
for (x_1^{**}, x_2^{**}), using e.g. Newton's method.

3. Update solution:
$$x_1^{k+1} = x_1^{**}, \qquad x_2^{k+1} = x_2^{**}.$$

4. Repeat steps 1-3 until convergence is reached:
$$\left|\left|R_1(x_1^{k+1}, x_2^{k+1})\right|\right|_\infty \leq \epsilon_1 \text{ and } \left|\left|R_2(x_1^{k+1}, x_2^{k+1})\right|\right|_\infty \leq \epsilon_2 \tag{3.4}$$

A flow chart of the method is shown on the left panel of Figure 3.1. This algorithm is the current approach to sequential-implicit simulations and the method described in Chapter 2 [28, 29, 30, 22, 43, 78, 55, 18]. The introduction of carefully chosen constraints $b(x_1, x_2)$ and $c(x_1, x_2)$ can enhance the convergence of the method. In domain decomposition methods, where the equations and unknowns are split across subdomain boundaries, the choice $b(x_1, x_2) = x_2$, $c(x_1, x_2) = x_1$ corresponds to a block Gauss-Seidel method, also known as a classical alternating Schwarz method with Dirichlet transmission conditions and minimal overlap [61]. Alternatively, if the constraints are chosen to match discrete Robin traces (i.e., a linear combination of function values and fluxes), then one obtains an optimized Schwarz method, which may converge a lot more quickly than block Gauss-Seidel [16, 17]. Note that although

(3.2) and (3.3) formally have the same dimensions as the fully coupled problem, the constraints $b(x_1, x_2)$ and $c(x_1, x_2)$ are generally chosen to be very simple (cf. $b(x_1, x_2) = x_2$, $c(x_1, x_2) = x_1$ for the block Gauss-Seidel case), so the suproblems in practice have much smaller effective sizes and are much cheaper to solve than the fully coupled problem. For multiphysics problems and as shown by the previous chapter, such constraints are often required for the convergence and stability [28, 30, 29, 78]. Through those studies, it was shown that a naive splitting of the subproblems and fixing the primary variables was insufficient for convergence and stability for these sequential schemes.

3.2.2 Sequential-implicit Newton method

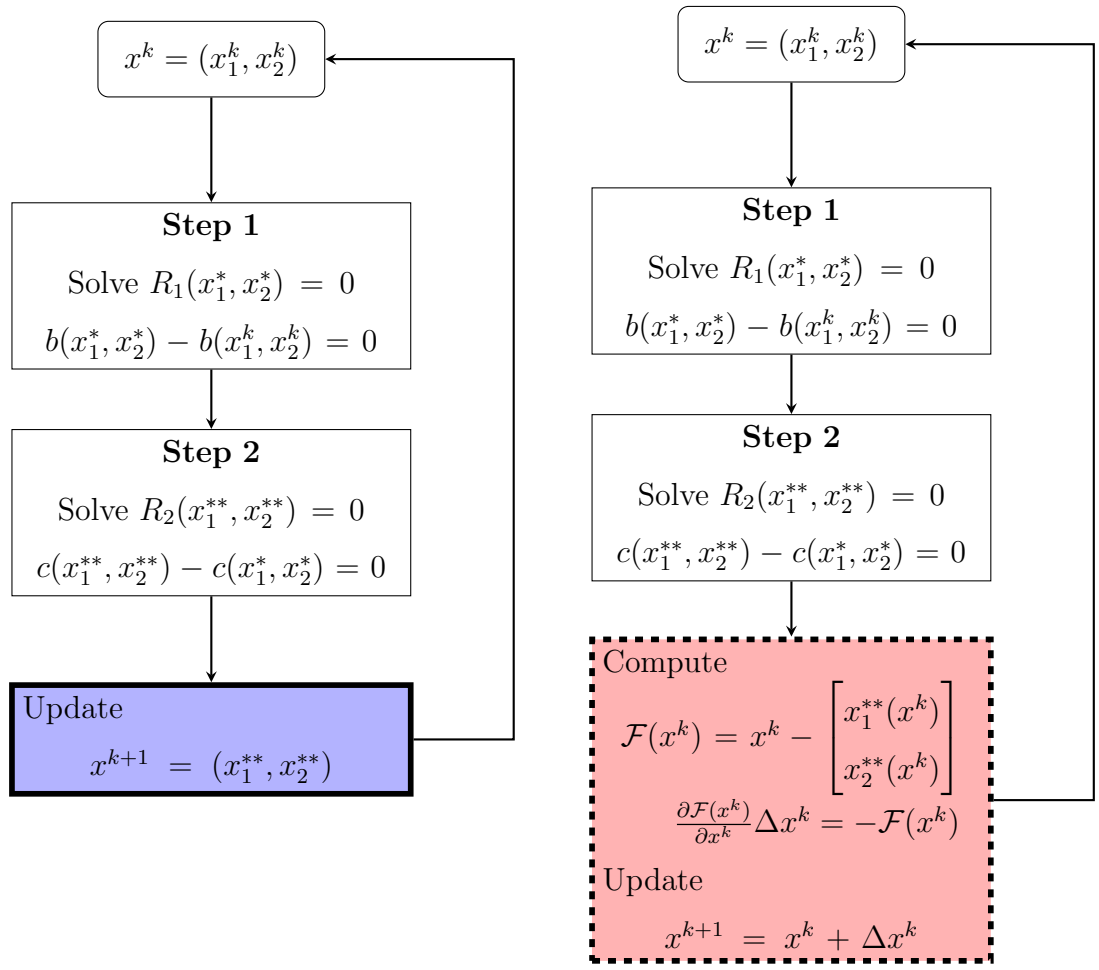
The key idea for accelerating the SIFP method is to note that at convergence, the fixed point of the SIFP method satisfies the equation:

$$x^{k+1} = (x_1^{**}(x^k), x_2^{**}(x^k)) = (G_1(x^k), G_2(x^k)) = G(x^k), \tag{3.5}$$

where $G_1(x^k)$ and $G_2(x^k)$ are the mappings defined by steps 1 and 2 in the SIFP method to obtain x_1^{**} and x_2^{**} based on the input (x_1^k, x_2^k). Thus, if we define the new function:

$$\mathcal{F}(x) = x - G(x), \tag{3.6}$$

then the fixed point of the SIFP method must be a solution of $\mathcal{F}(x) = 0$. The Sequential-implicit Newton (SIN) method consists of solving this equation using Newton's method.



(a) Sequential-implicit fixed-point (SIFP) (b) Sequential-implicit Newton (SIN)

Figure 3.1: Flowchart of sequential scheme using a fixed-point iteration update (Left) and a Newton update (Right)

A flowchart of this algorithm is presented in the right panel of Figure 3.1. The key difference with SIFP is how the solution is updated after all the subproblems are solved: SIFP takes directly the subproblem solutions $(x_1^{**}, x_2^{**}) = (G_1(x^k), G_2(x^k))$ as the new iterate, whereas SIN performs a Newton update by linearizing the fixed point function $\mathcal{F}$ at x^k. The added complexity for SIN is that the Jacobian for the nonlinear function $\mathcal{F}$ is required. However, rather than computing an explicit representation of $\frac{\partial \mathcal{F}(x)}{\partial x}$, we exploit the fact the Jacobian $\frac{\partial \mathcal{F}(x)}{\partial x}$ is only required when we solve the

system of linear equations

$$\frac{\partial \mathcal{F}(x^k)}{\partial x}\Delta x^k = -\mathcal{F}(x^k). \tag{3.7}$$

Hence, if we have a routine for calculating the matrix-vector product for the Jacobian $\left(\frac{\partial \mathcal{F}(x^k)}{\partial x}\right)$, then we can use a Krylov solver such as GMRES [57] to solve the linear equations, thus avoiding the explicit calculation and storage of the Jacobian.

We will now explain how the Jacobian matrix-vector product for the SIN update can be computed. For notational convenience, let us rewrite x_1^* and x_2^* calculated in Step 1 of SIFP as functions of the input arguments $x = (x_1, x_2)$, i.e., we define:

$$x_1^* = H_1(x), \qquad x_2^* = H_2(x).$$

Then H_1 and H_2 satisfy:

$$R_1(H_1(x), H_2(x)) = 0 \tag{3.8}$$

and also the constraint:

$$b(H_1(x), H_2(x)) = b(\Pi_1 x, \Pi_2 x), \tag{3.9}$$

where Π_1 and Π_2 are projections that restrict the solution vector to its first and second block components respectively, i.e., $\Pi_1 x = x_1$ and $\Pi_2 x = x_2$. Now Step 2 of the algorithm can be written as:

$$R_2(G_1(x), G_2(x)) = 0, \tag{3.10}$$

$$c(G_1(x), G_2(x)) = c(H_1(x), H_2(x)). \tag{3.11}$$

We wish to solve the equation $\mathcal{F}(x) = x - G(x) = 0$ using Newton's method, which involves repeatedly solving linear systems of the type (3.7). Solving this linear system

GMRES requires us to derive an expression for $\left(\frac{\partial \mathcal{F}}{\partial x}\right) v$, where v is an arbitrary vector. This can be done using implicit differentiation. Differentiating Equations (3.8), (3.9) with respect to x leads to:

$$\frac{\partial R_1}{\partial x_1}(H_1(x), H_2(x))\frac{\partial H_1}{\partial x} + \frac{\partial R_1}{\partial x_2}(H_1(x), H_2(x))\frac{\partial H_2}{\partial x} = 0, \tag{3.12}$$

$$\frac{\partial b}{\partial x_1}(H_1(x), H_2(x))\frac{\partial H_1}{\partial x} + \frac{\partial b}{\partial x_2}(H_1(x), H_2(x))\frac{\partial H_2}{\partial x} = \frac{\partial b}{\partial x_1}(\Pi_1 x, \Pi_2 x)\Pi_1 + \frac{\partial b}{\partial x_2}(\Pi_1 x, \Pi_2 x)\Pi_2. \tag{3.13}$$

Multiplying Equations (3.12) and (3.13) from the right by the arbitrary vector v and noting that $\Pi_i x = x_i$, $\Pi_i v = v_i$ for $i = 1, 2$, we can rewrite the result as a linear system of the form:

$$\begin{bmatrix} \frac{\partial R_1}{\partial x_1}(H_1(x), H_2(x)) & \frac{\partial R_1}{\partial x_2}(H_1(x), H_2(x)) \\ \frac{\partial b}{\partial x_1}(H_1(x), H_2(x)) & \frac{\partial b}{\partial x_2}(H_1(x), H_2(x)) \end{bmatrix} \begin{bmatrix} \frac{\partial H_1}{\partial x} v \\ \frac{\partial H_2}{\partial x} v \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial b}{\partial x_1}(x_1, x_2)v_1 + \frac{\partial b}{\partial x_2}(x_1, x_2)v_2 \end{bmatrix}. \tag{3.14}$$

Thus, $(\frac{\partial H_1}{\partial x} v, \frac{\partial H_2}{\partial x} v)$ can be computed by solving the linear system (3.14). Note that the matrix on the left-hand side is simply the Jacobian of the nonlinear subproblem in Step 1 of SIFP, evaluated at the solution $(H_1(x), H_2(x)) = (x_1^*, x_2^*)$. If Newton's method is used to solve the subproblem, the LU factors of this Jacobian would already have been computed, so they can be reused to solve (3.14). Also note that the derivatives of the constraints $\frac{\partial b}{\partial x_j}$ appear on both sides of the equation, but they are evaluated at different arguments (at the solution (x_1^*, x_2^*) on the left, and at the inputs (x_1, x_2) on the right). Similarly, we can apply implicit differentiation to Equations (3.10), (3.11) to obtain:

$$\begin{bmatrix} \frac{\partial R_2}{\partial x_1}(G_1(x), G_2(x)) & \frac{\partial R_2}{\partial x_2}(G_1(x), G_2(x)) \\ \frac{\partial c}{\partial x_1}(G_1(x), G_2(x)) & \frac{\partial c}{\partial x_2}(G_1(x), G_2(x)) \end{bmatrix} \begin{bmatrix} \frac{\partial G_1}{\partial x} v \\ \frac{\partial G_2}{\partial x} v \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\partial c}{\partial x_1}(x_1^*, x_2^*)\frac{\partial H_1}{\partial x} v + \frac{\partial c}{\partial x_2}(x_1^*, x_2^*)\frac{\partial H_2}{\partial x} v \end{bmatrix}, \tag{3.15}$$

where the right hand side contains the previously calculated terms $\frac{\partial H_1}{\partial x}v$ and $\frac{\partial H_2}{\partial x}v$. Again, the left-hand side matrix is simply the Jacobian of the nonlinear subproblem in Step 2, evaluated at the solution (x_1^{**}, x_2^{**}), so reusing the LU factors is possible. Finally, we can compute the matrix-vector product for an arbitrary vector v via the formula:

$$\frac{\partial \mathcal{F}}{\partial x}v = v - \begin{bmatrix} \frac{\partial G_1}{\partial x}v \\ \frac{\partial G_2}{\partial x}v \end{bmatrix}, \tag{3.16}$$

where $\frac{\partial G_1}{\partial x}v$ and $\frac{\partial G_2}{\partial x}v$ are calculated by solving (3.15).

We summarize the full SIN algorithm below. For $k = 0, 1, 2, \ldots$, proceed as follows:

1. Solve:

$$R_1(x_1^*, x_2^*) = 0, \qquad b(x_1^*, x_2^*) = b(x_1^k, x_2^k)$$

for (x_1^*, x_2^*). At convergence, store the most recent Jacobian matrix $\begin{bmatrix} \frac{\partial R_1}{\partial x_1} & \frac{\partial R_1}{\partial x_2} \\ \frac{\partial b}{\partial x_1} & \frac{\partial b}{\partial x_2} \end{bmatrix}$, evaluated at (x_1^*, x_2^*), and its LU factors for use in Step 4.

2. Solve:

$$R_2(x_1^{**}, x_2^{**}) = 0, \qquad c(x_1^{**}, x_2^{**}) = c(x_1^*, x_2^*)$$

for (x_1^{**}, x_2^{**}). At convergence, store the most recent Jacobian matrix $\begin{bmatrix} \frac{\partial R_2}{\partial x_1} & \frac{\partial R_2}{\partial x_2} \\ \frac{\partial c}{\partial x_1} & \frac{\partial c}{\partial x_2} \end{bmatrix}$, evaluated at (x_1^{**}, x_2^{**}), and its LU factors for use in Step 4.

3. Compute the residual:

$$r^k := \mathcal{F}(x^k) = \begin{bmatrix} x_1^k - x_1^{**} \\ x_2^k - x_2^{**} \end{bmatrix}. \tag{3.17}$$

4. Solve:

$$\frac{\partial \mathcal{F}}{\partial x}\Delta x^k = -r^k \tag{3.18}$$

for Δx^k using a Krylov solver such as GMRES. To multiply $\frac{\partial \mathcal{F}}{\partial x}$ by an arbitrary vector v, first solve (3.14) for $(\frac{\partial H_1}{\partial x}v, \frac{\partial H_2}{\partial x}v)$, then solve (3.15) for $(\frac{\partial G_1}{\partial x}v, \frac{\partial G_2}{\partial x}v)$, and finally compute $\frac{\partial \mathcal{F}}{\partial x}v$ via (3.16).

5. Update the solution by setting:

$$x^{k+1} = x^k + \Delta x^k.$$

6. Repeat Steps 1–5 until convergence.

3.2.3 Computational cost of SIFP vs SIN

We now show a theoretical comparison of the cost of the SIFP and SIN methods, based on their local convergence properties and the computational cost of each step. We first define a number of parameters needed for estimating the running time. In Steps 1 and 2, which are common for both methods, suppose we solve the nonlinear sequential subproblems using Newton's method. This requires a number of inner Newton iterations per nonlinear solve, which we denote by K_{Newton}. Typically, this number should be somewhere between 3 and 10, but it can be higher for difficult problems. (Because the sequential subproblems are similar for the two methods, we will use the same K_{Newton} for both.) Within each Newton iteration, we need to calculate the LU factors of the Jacobian matrix, and then we need to perform forward and backward substitution to solve the associated systems. Let T_{fact} and T_{sub} be the corresponding running times (where typically T_{fact} is between one and two orders of magnitude larger than T_{sub}, see for instance [35, 36]). In addition, Step 4 of SIN requires the solution of a linear system by GMRES, where each GMRES

iteration requires solving (but not factoring) a linear system containing the Jacobian of the sequential subproblems. Assuming that K_{GMRES} such iterations are needed, we deduce that the total running times for SIFP and SIN are given by:

$$T_{\text{SIFP}} = K_{\text{SIFP}} K_{\text{Newton}} (T_{\text{fact}} + T_{\text{sub}}), \tag{3.19}$$

$$T_{\text{SIN}} = K_{\text{SIN}} (K_{\text{Newton}} (T_{\text{fact}} + T_{\text{sub}}) + K_{\text{GMRES}} T_{\text{sub}}), \tag{3.20}$$

where K_{SIFP} and K_{SIN} are the number of outer SIFP and SIN iterations required for convergence.

We now compare the various quantities above. First, we compare K_{SIFP} and K_{SIN}. Observe that the SIFP method is a fixed point method of the type $x^{k+1} = G(x^k)$. If the method is locally convergent near the fixed point $\bar{x} = G(\bar{x})$, then it is well known [47] that the asymptotic convergence rate, which is defined as:

$$\rho := \limsup_{k \to \infty} \|x^k - \bar{x}\|,$$

is given by the spectral radius of the Jacobian matrix of G evaluated at the fixed point. In other words, we have:

$$\rho = \max_j |\lambda_j|,$$

where the λ_j are the eigenvalues of $G^* := \frac{\partial G}{\partial x}(\bar{x})$. Thus, unless all the eigenvalues of the Jacobian are zero (i.e., the matrix is nilpotent), we can only expect the SIFP method to converge linearly in a neighborhood of the fixed point. In contrast, SIN is a Newton method applied to the nonlinear equation $\mathcal{F}(x) = 0$. Because the exact Jacobian is used to calculate the Newton update at every iteration, we expect the method to converge quadratically close to the solution. Thus, we expect $K_{\text{SIFP}} \gg K_{\text{SIN}}$.

Next, we show the estimate of K_{GMRES}. To do so, we need to consider the properties of the Jacobian matrix $\frac{\partial \mathcal{F}}{\partial x}(\bar{x})$ close to the fixed point. Recall that for a general nonsingular linear system $Au = b$, GMRES finds in k iterations the solution u^k that minimizes the 2-norm of the residual $r^k = b - Au^k$. Equivalently, for a given initial residual $r^0 = b - Au^0$, GMRES chooses the best polynomial $p(z)$ of degree k or lower, such that $p_k(0) = 1$ and $r^k = p_k(A)r^0$ is minimized [58]. Thus, for any other degree k polynomial q_k with $q_k(0) = 1$, we necessarily have:

$$\|r^k\|_2 = \|p_k(A)r^0\|_2 \leq \|q_k(A)r^0\|_2 \leq \|q_k(A)\|_2\|r^0\|_2.$$

This property allows us estimate the convergence rate of GMRES by guessing a polynomial $q_k(z)$, knowing that the true residual must be smaller. In the case of SIN, when x^k is close to $\bar{x}$, we have:

$$\frac{\partial \mathcal{F}}{\partial x}(x^k) \approx \frac{\partial \mathcal{F}}{\partial x}(\bar{x}) = I - G^*.$$

Thus, by choosing $q_k(z) = (1 - z)^k$, we see that $q_k(0) = 1$ and $q_k(I - G^*) = (G^*)^k$. Thus, we have:

$$\|r^k\|_2 \leq \|q_k(I - G^*)r^0\|_2 = \|(G^*)^k r^0\|_2 \leq \|(G^*)^k\|_2\|r^0|_2 \leq C\rho^k\|r^0\|_2.$$

Thus, GMRES converges at an asymptotic rate that is at least as good as ρ, so we have $K_{\text{GMRES}} \lesssim K_{\text{SIFP}}$. In practice, our numerical experiments show that K_{GMRES} is in fact much smaller than K_{SIFP}, see Sections 3.5 and 3.4.

Referring back to Equations (3.19) and (3.20), we see that the running time of SIN is lower than that of SIFP if:

$$K_{\text{SIN}}K_{\text{GMRES}}T_{\text{sub}} \leq (K_{\text{SIFP}} - K_{\text{SIN}})K_{\text{Newton}}(T_{\text{fact}} + T_{\text{sub}}). \tag{3.21}$$

Using the fact that $K_{\text{GMRES}} \lesssim K_{SIFP} - K_{SIN}$, that $T_{\text{fact}} \gg T_{\text{sub}}$ and that $K_{\text{SIN}} \approx K_{\text{Newton}}$ in most cases (since both are Newton iterations and converge quadratically locally), we see that (3.21) holds in the vast majority of cases, meaning that we expect SIN require less computation than SIFP in most cases. Our numerical experiments shown in the next sections confirmed that this is indeed the case for the two problem classes that we consider.

3.3 AD-GPRS

3.3.1 General Sequential Framework

The SIN method was implemented into the Automatic-Differentiation General Purpose Research Simulator (AD-GPRS). AD-GPRS consists of a general sequential-implicit coupling framework for solving multiphysics problems for reservoir simulation [55]. This framework allowed for the consistent testing and development of the SIN and the SIFP methods. The framework employs a modular code design by splitting each individual physics into a set of different subproblems. The main components of this framework use a subproblem tree structure and abstract computational domains to separate and organize the variable sets for each subproblem. This allows for minimal code duplication and ensures for consistent comparisons between formulations and algorithms.

To activate the sequential-implicit Newton option, rather than specifying the SEQ keyword for the SIFP or fully coupled method, a SIN keyword is used (Figure 3.2). The SIN keyword follows the same structure as the SEQ keyword where you are able to specify the number of outer loop iterations for the sequential scheme. Currently, the SIN keyword only supports two different subproblems rather than the fully general case available with SEQ where an arbitrary number of subproblems is possible.

```
COUPLING
SIN(R1,R2,30)
/
```

Figure 3.2: AD-GPRS Input File Example for Sequential Newton

One of the key steps in the SIN method is that the Jacobians (J_{11}, J_{12}, J_{21} and J_{22}) are required after steps 2 and 3. To compute and store these Jacobians, the sequential framework is exploited. Here we add two additional FIM steps to compute the Jacobians necessary. These additional sequential steps are shown in Figure 3.3. Thus rather than changing the sequential framework structure, we implement this equivalent sequential structure if the SIN keyword is activated.

```
COUPLING
SEQ(R1<MAP_FULL>,FIM(R1<MAP_FULL>,R2<NO_RESIDUAL>),
      R2<MAP_FULL>,FIM(R1<NO_RESIDUAL>,R2<MAP_FULL>,30)  /
/
```

Figure 3.3: AD-GPRS Equivalent Input File for Sequential Newton

There are four different nodes within this sequential tree structure. Each of these nodes have a specific purpose in the SIN computation:

1. R1<MAP_FULL> - this step solves the first residual equation corresponding to step 2 in the full algorithm
2. FIM(R1<MAP_FULL>,R2<NO_RESIDUAL>) - this computes and stores J_{11}, J_{12}
3. R2<MAP_FULL> this step solves the second residual equation, corresponding to step 3 in the full algorithm

4. FIM(R1<NO_RESIDUAL>,R2<MAP_FULL>) - this computes and stores J_{21}, J_{22}

Constrained Variable Derivatives

The Jacobian matrices associated to the different constrained variables are also required. To achieve this, we introduce a new array called vConstrainedVariables. This contains all the constrained variables for the different multiphysics coupling strategies. Currently, there are two ConstrainedVariable objects that are stored in this array. This follows a very similar structure to the CoupledProperties object introduced in the GENIC framework [55], where we call the compute function after FIM(R1<MAP_FULL>,R2<NO_RESIDUAL>) and FIM(R1<NO_RESIDUAL>, R2<MAP_FULL>) sequential solves. There are two main methods within the ConstrainedVariable object:

1. isActive ())

   This step specifies whether or not the constrained variable is active. This allows the ConstrainedVariable to distinguish whether we require the derivatives for b or c. This indicates whether it is at step 2 or 3 in the solution process.

2. compute(Vector &deriv_dx1, Vector &deriv_dx2)

   This computes the derivatives associated to b_{x_1}, b_{x_2}, c_{x_1}, c_{x_2} in steps 2 and 3. This function is called after the sequential nodes:

   - FIM(R1<MAP_FULL>,R2<NO_RESIDUAL>)
   - FIM(R1<NO_RESIDUAL>,R2<MAP_FULL>)

   This ensures that it is consistent with the Jacobian calculations (J_{11}, J_{12}, J_{21}, J_{22})

Two ConstrainedVariable child classes GeothermalCV and GeomechanicsCV were implemented. They each represent the two SIN formulations investigated in this research: flow-thermal and flow-mechanics.

For GeothermalCV, we compute the derivative for c depending on the constraint strategy used: fixed pressure, fixed density or hybrid. As this constraint is applied locally on the block, we store the Jacobian c_{x_1} and c_{x_2} as a vector representing the diagonal of the Jacobian.

For GeomechanicsCV, this is slightly different due to how the fixed stress constraint is applied. Here the derivative that is computed is J_σ, which is a diagonal Jacobian representing how the flow residual changes with the stress of the problem. This is diagonal as the stress only modifies the porosity of the problem. If there is a permeability modification, this method will have to be changed to account for the inter-block connections.

3.3.2 Linear System Structure

The main difference between the SIN and SIFP method is how each sequential update is calculated. One of the key distinctions between the SIN and SIFP method is that additional Jacobian matrices are required to compute the sequential Newton update, these additional steps are shown in Figure 3.3. In order to compute the Newton update, Equation (3.18) is solved. However we note that we do not need an explicit computation of the Jacobian but only the matrix-vector product of the Jacobian $\left(\frac{\partial \mathcal{F}}{\partial x}\right)$ is required. To solve the system of equations we use a templated implementation of GMRES described in page 20 of Barrett et al. [4]. We do not use a preconditioner for this GMRES solve. The main operation that needs to be defined is how the matrix multiplication is performed. Here define a new class NewtonSeqMatrix that is a child class of the GPRS_Matrix class. This inherits all the basic matrix operators. The only pure virtual function in GPRS_Matrix that must be implemented is axpy (axpy: y = y + A $*$ v). To implement this, we follow the steps listed in Steps 5a and 5b, to obtain this matrix-vector multiplication.

3.4 Flow-Thermal Problem

In this section, we demonstrate the effectiveness of the SIN method for the geothermal problem as described in chapter 2. This geothermal problem involves the flow and transport of pure water in two phases with thermal effects. The flow equation takes the form:

$$\frac{\partial}{\partial t}\left(\phi \sum_{l=1}^{2} \rho_l S_l\right) - \nabla.\left(\sum_{l=1}^{2} (\rho_l v_l)\right) - Q_M = 0, \tag{3.22}$$

where:

- ϕ is the porosity of the rock;
- ρ_l is the mass density of phase l;
- S_l is the saturation of phase l;
- $\boldsymbol{v}_l$ is the velocity of the phase l;
- Q_M is the mass source/sink term.

The subscripts l represents the phase of the fluid. The density ρ_l of each phase depends on the phase state of the fluid and is a function of pressure and enthalpy. In addition to these the conservation equations, the saturation constraint must be satisfied; that is, the sum of all the phase saturation is unity:

$$\sum_{l=1}^{2} S_l = 1 \tag{3.23}$$

The thermal residual equation involves solving the energy conservation equation:

$$\frac{\partial}{\partial t}\left[(1-\phi)\,\rho_R U_R + \phi \sum_{l=1}^{2} \rho_l U_l S_l\right] - \nabla.\left(\sum_{l=1}^{2} (\rho_l h_l u_l)\right) - \nabla.\,(K\nabla T) - Q_E = 0 \tag{3.24}$$

where:

- Q_E is the energy source/sink term
- h_l is the phase enthalpy of phase l
- U_l is the internal energy of phase l
- K is the total thermal conductivity of the fluids and rock.

The mass density ρ_l and enthalpy h_l of each phase depends on the phase state of the fluid. For single-phase conditions, ρ_l and h_l are functions of pressure and temperature. The thermodynamic relationships of water are taken from Faust and Mercer [15]. However, for two-phase conditions, ρ_l and h_l will depend only on the pressure because now pressure and temperature are dependent: we have $p = p_{sat}(T)$, where p_{sat} is the saturated pressure as a function of temperature. To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium:

$$\boldsymbol{v}_l = -\frac{kk_{rl}}{\mu_l}\nabla(p_l + \rho_l g z) \tag{3.25}$$

where:

- $\boldsymbol{v}_l$ is the superficial velocity of the phase l;
- k is the rock permeability;
- k_{rl} is the relative permeability for phase l;
- p_l is the pressure of phase l (here we neglect any capillary pressure effects, so the pressure of the phases are equal);
- g is the gravitational constant;
- μ_l is the viscosity of the phase l;
- z is the coordinate direction of gravity.

3.4.1 Fully Coupled Formulation

For the fully coupled solution, we discretized the mass and energy conservation equations in space using the two-point flux approximation (TPFA) finite volume method, with single-point upstream weighted scheme for the flux discretization. We used a pressure-enthalpy formulation as the primary variables [77]. After discretization, we obtain a discrete algebraic problem of the form:

$$\begin{cases} R_F(p_{n+1}, h_{n+1}) = 0, \\ R_T(p_{n+1}, h_{n+1}) = 0, \end{cases} \tag{3.26}$$

where:

- $p_{n+1}, h_{n+1} \in \mathbb{R}^{N_c}$ is the vector of pressures and enthalpies at each of the N_c cell centers at time t_{n+1};
- $R_F : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the mass conservation equation (3.22), as a function of pressure and enthalpy;
- $R_T : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the energy balance equations (3.24), as a function of pressure and enthalpy.

At the start of each timestep, we use the solution of the previous time step (p_n, h_n) as the initial guess. Thus, fully coupled iterative process for the flow-thermal problem is:

1. Solve for $p_{n+1}^{k+1}, h_{n+1}^{k+1}$ for:

$$\begin{bmatrix} \frac{\partial R_F}{\partial p} & \frac{\partial R_F}{\partial h} \\ \frac{\partial R_T}{\partial p} & \frac{\partial R_T}{\partial h} \end{bmatrix}_{n+1}^{k} \begin{bmatrix} \Delta p^k \\ \Delta h^k \end{bmatrix} = \begin{bmatrix} R_F(p_{n+1}^k, h_{n+1}^k) \\ R_T(p_{n+1}^k, h_{n+1}^k) \end{bmatrix} \tag{3.27}$$

   where

- $\Delta p^k = p_{n+1}^{k+1} - p_{n+1}^k$
- $\Delta h^k = h_{n+1}^{k+1} - h_{n+1}^k$
- The Jacobian matrices $\frac{\partial R_F}{\partial p}$, $\frac{\partial R_F}{\partial h}$, $\frac{\partial R_T}{\partial p}$, $\frac{\partial R_T}{\partial h}$ are all evaluated at (p_{n+1}^k, h_{n+1}^k)

2. Step 1 is repeated until convergence:

$$\left|\left|R_F(p_{n+1}^{k+1}, h_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_p \quad \text{and} \quad \left|\left|R_T(p_{n+1}^{k+1}, h_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_T. \tag{3.28}$$

3.4.2 Sequential Formulation

We will apply the general formulation described in Section 3.2.1 for the flow and thermal problem and describe the solution process for solving each of the steps in terms of the specific primary variables and constraints applied for the flow and thermal problem.

At the start of each timestep, we use the solution of the previous timestep as the initial guess $x_{n+1}^0 = (p_{n+1}^0, h_{n+1}^0) = (p_n, h_n) = x_n$. Where $p_{n+1}^0 = p^0 \in \mathbb{R}^{N_c}$ and $h_{n+1}^0 = h^0 \in \mathbb{R}^{N_c}$ are the pressure and enthalpy at the cell centers (N_c is the number of cells). For simplicity, we will now drop the subscript $n+1$ as all following terms represent the solution for the $n+1$ timestep. The sequential iterative process for the flow-thermal problem is:

1. Solve for p^* where $R_F(p^*, h^k)$=0, the constraint for his case is $h^* = h^k$, so we do not need to compute the Jacobian for the b constraint. Convergence is defined as $\left|\left|R_F(p^*, h^k)\right|\right|_\infty \leq \epsilon_F$;

2. Solve for p^{**}, h^{**} where $R_T(p^{**}, h^{**}) = 0$, while satisfying $c(p^*, h^k) = c(p^{**}, h^{**})$, here c is the constraint employed. Convergence is defined as:

$$||R_T(p^{**}, h^{**})||_\infty \leq \epsilon_T \quad \text{and} \quad \left|\left|c(p^*, h^0) - c(p^{**}, h^{**})\right|\right|_\infty \leq \epsilon_T;$$

3. Update x^{k+1} by fixed-point iteration or Newton's method;

4. Repeat steps 1-3 until convergence:

$$\left|\left|R_F(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_F \quad \text{and} \quad \left|\left|R_T(p^{k+1}, h^{k+1})\right|\right|_\infty \leq \epsilon_T. \tag{3.29}$$

3.4.3 Constraints

In this study we investigated three different constraints: fixed pressure, fixed density and a hybrid approach. These are the same constraints described in Chapter 2, but we will restate them here for completeness. All three constraints are applied in the second step when the thermal residual is solved.

Fixed Pressure

The fixed pressure scheme assumes that the pressure at each cell is fixed when solving for the thermal residual $c := p$. So $p^{**} = p^*$:

$$R_T(p^{**}, h^{**}) = 0 \tag{3.30}$$

$$p^{**} - p^* = 0 \tag{3.31}$$

Fixed Density

The fixed density scheme assumes that the density at each cell is fixed when solving for the thermal residual $c := \rho$. So $\rho^* = \rho(p^*, h^k)$, where ρ is a function that computes the cell center densities based on the pressure and enthalpy of each cell. ($\rho : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \rightarrow \mathbb{R}^{N_c}$)

$$R_T(p^{**}, h^{**}) = 0 \tag{3.32}$$

$$\rho(p^{**}, h^{**}) - \rho(p^{*}, h^{k}) = 0 \tag{3.33}$$

Fixed Pressure and Fixed Density (Hybrid)

The hybrid scheme combines the fixed pressure and fixed density constraints. This constraint varies cell-wise and is based on the phase state of the cell. A fixed pressure is enforced for single-phase cells and fixed density for two-phase cells. We determine the phase state of the cell by comparing the enthalpy with the saturated enthalpy of the cell, $h_w(p)$ and $h_s(p)$ are the saturated water and steam enthalpy respectively. The constraint for each cell i with pressure p_i and and enthalpy h_i:

$$c(p_i, h_i) := \begin{cases} \rho(p_i, h_i) & h_w(p_i) \leq h_i \leq h_s(p_i) \text{ (Two-phase)} \\ p_i & \text{otherwise (Single-phase)} \end{cases} \tag{3.34}$$

3.4.4 Sequential-implicit Fixed Point Algorithm

The update for the SIFP fixed stress algorithm is simply using the solutions from the flow and thermal residual equations as the update:

$$x^{k+1} = (p^{**}, h^{**}) \tag{3.35}$$

3.4.5 Sequential-implicit Newton Algorithm

The primary operation required to obtain an update for the SIN method is to compute the matrix-vector product for the Jacobian $\frac{\partial \mathcal{F}}{\partial x}$, where $\mathcal{F}$ is the preconditioned system defined in Equation 3.36. For the flow and thermal problem this preconditioned system is:

$$\mathcal{F}(p^k, h^k) = \begin{bmatrix} p^k - p^{**}(p^k, h^k) \\ h^k - h^{**}(p^k, h^k) \end{bmatrix} \tag{3.36}$$

The steps to compute the update Δx^k for a given iteration p^k and h^k are as follows:

1. Calculate p^*, by solving $R_F(p^*, h^k) = 0$ using Newton's method, at convergence ($\left|\left|R_F(p^*, h^k)\right|\right|_\infty \leq \epsilon_F$) store:
   - $J_{11} = \frac{\partial R_F}{\partial p}\Big|_{\partial h=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $J_{12} = \frac{\partial R_F}{\partial h}\big|_{\partial p=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $c_p^0 = \frac{\partial c}{\partial p}\Big|_{\partial h=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
   - $c_h^0 = \frac{\partial c}{\partial h}\big|_{\partial p=0}(p^*, h^k) \in \mathbb{R}^{N_c \times N_c}$
2. Calculate p^{**}, h^{**}, by solving $R_T(p^{**}, h^{**}) = 0$,$c(p^*, h^0) - c(p^{**}, h^{**}) = 0$, at convergence ($||R_T(p^{**}, h^{**})||_\infty \leq \epsilon_T$ and $||c(p^{**}, h^{**}) - c(p^*, h^*)||_\infty \leq \epsilon_T$) store:
   - $J_{21} = \frac{\partial R_T}{\partial p}\Big|_{\partial h=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
   - $J_{22} = \frac{\partial R_T}{\partial h}\big|_{\partial p=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$ or store the LU factors used to multiply by $\left(J_{22} - J_{21}(c_p^1)^{-1}c_h^1\right)^{-1}$ that is computed when solving $R_T(p^{**}, h^{**}) = 0$,$c(p^*, h^0) - c(p^{**}, h^{**}) = 0$
   - $c_p^1 = \frac{\partial c}{\partial p}\Big|_{\partial h=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
   - $c_h^1 = \frac{\partial c}{\partial h}\big|_{\partial p=0}(p^{**}, h^{**}) \in \mathbb{R}^{N_c \times N_c}$
3. Calculate:
$$r^k := \mathcal{F}(x^k) = \begin{bmatrix} p^k - p^{**} \\ h^k - h^{**} \end{bmatrix} \in \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \tag{3.37}$$
4. Solve:
$$\frac{\partial \mathcal{F}}{\partial x}\Delta x^k = -r^k \tag{3.38}$$
for Δx^k using GMRES. This will require multiplying $\frac{\partial \mathcal{F}}{\partial x}$ by an arbitrary vector

$v = (v_1, v_2)$, where:

$$\frac{\partial \mathcal{F}}{\partial x} v = v - \begin{bmatrix} \frac{\partial G_1}{\partial x} \\ \frac{\partial G_2}{\partial x} \end{bmatrix} v = \begin{bmatrix} v_1 - w_1 \\ v_2 - w_2 \end{bmatrix} \tag{3.39}$$

(a) Compute $z_1 = -J_{11} \left(J_{12} v_2 \right) \in \mathbb{R}^{N_c}$

(b) Compute:

$$w_2 = \left(J_{22} - J_{21} \left(c_p^1 \right)^{-1} c_h^1 \right)^{-1} \left(-J_{21} (c_p^1)^{-1} \left[c_p^0 z_1 + c_h^0 v_2 \right] \right) \in \mathbb{R}^{N_c} \tag{3.40}$$

Here we compute the Schur complement of the large block matrix system in Equation 3.15. The two inverses $\left(c_p^1 \right)^{-1}$ and $\left(c_p^1 \right)^{-1}$ are cheap to calculate since they are both diagonal matrices. The matrix multiplication by $\left(J_{22} - J_{21} (c_p^1)^{-1} c_h^1 \right)^{-1}$ could also utilize the LU factors at step 2 if a direct solver was used.

(c) Compute:

$$w_1 = \left(c_p^1 \right)^{-1} \left(c_p^0 z_1 + c_h^0 h - c_h^1 w_2 \right) \tag{3.41}$$

5. Update:

$$\begin{bmatrix} p^{k+1} \\ h^{k+1} \end{bmatrix} = \Delta x^k + \begin{bmatrix} p^k \\ h^k \end{bmatrix} \tag{3.42}$$

6. Repeat steps 1-5 until:

$$\left|\left| [R_F(p^{k+1}, h^{k+1}) \right|\right|_\infty \leq \epsilon_F \text{ and } \left|\left| [R_T(p^{k+1}, h^{k+1}) \right|\right|_\infty \leq \epsilon_T \tag{3.43}$$

One-dimensional radial example

This numerical model is a one-dimensional, radial example. Here we looked at single- and two-phase flow regimes. The single-phase case involves a cold water that is

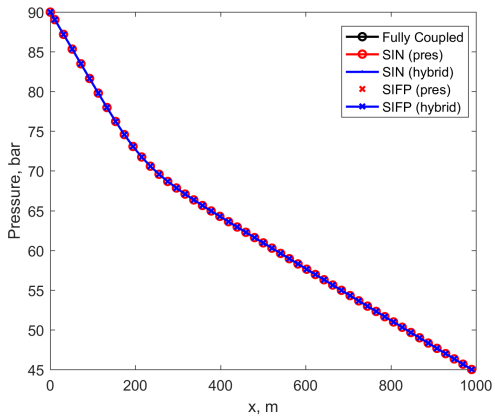
injected into a hot water reservoir. For the two-phase case, cold water is injected into a two-phase reservoir at saturated conditions. This model is based on the one-dimensional model investigated in the Stanford Code Comparison study [62].

The rock parameters for this model are:

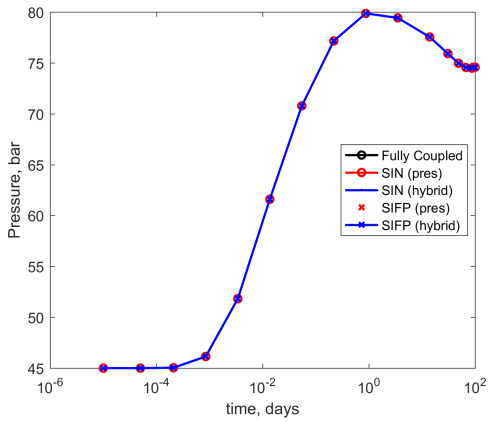
Table 3.1: Rock parameters

Property	**Value**	**Unit**
Reservoir Length	1000	m
Reservoir Thickness	100	m
Permeability	100	md
Rock density	2500	kg/m3
Rock specific heat capacity	1.0	J/(gK)
Porosity	0.2	%

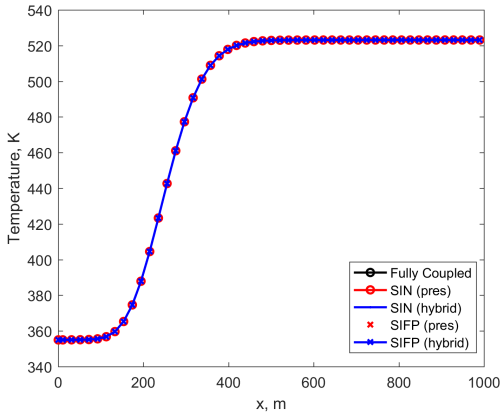
In this study we looked at a 50 cell one-dimensional radial model. Here we used $\epsilon_F = \epsilon_T = 10^{-4}$. The tolerance for the SIN GMRES solution step was set to 10^{-8}. The linear solver used to multiply $\left(J_{22} - J_{21}(c_p^1)^{-1}c_h^1\right)^{-1}$ was SuperLU [36], this was to decouple any effects the linear solver would have on the sequential updates.



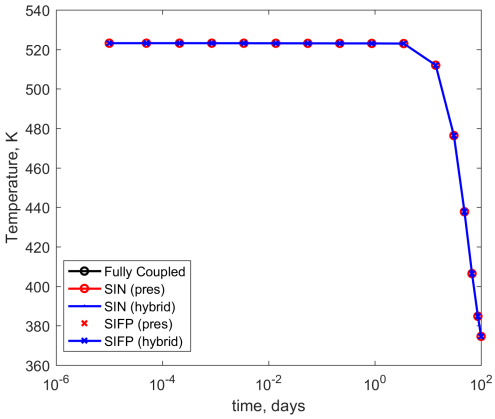
(a) Pressure profile at final time step



(b) Pressure over time at x = 167



(c) Temperature profile at final time step



(d) Temperature over time at x = 167

Figure 3.4: The pressure and temperature profiles for 50 cell single-phase injection

Figure 3.4 shows that the results of both SIN and SIFP are in close agreement with the fully coupled method. This is consistent with the results shown by [37] that they should both converge to the same solution. Table 3.2 shows the nonlinear results for the different methods. The "Full Newton iterations" row represents the number of Newton iterations that the fully coupled problem required over the entire simulation. The "Inner Newton iterations" row is the number of smaller inner loop Newton iterations performed. The "Sequential outer iterations" are the number of

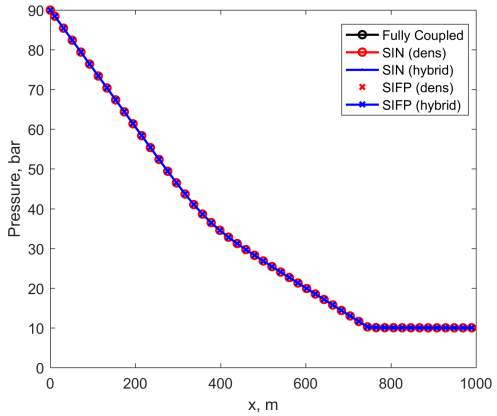
outer loop iterations to converge to the solution of the full problem. The "GMRES iterations" is the total number of GMRES iterations to solve Equation (3.38). The "Wasted timesteps" is the number of timesteps that did not converge either due to an unphysical update, an inner Newton loop not converging or the maximum number of sequential iterations was reached. The "Wasted Full Newtons" are the number of full Newton iterations that were computed for those timesteps that were wasted, this is only applicable for the fully coupled method, because the SIFP and SIN only run the smaller inner Newtons. The "Wasted Inner Newtons" are the number of Newton iterations computed for the timesteps that were wasted, this only applies for SIFP and SIN. "Newton/Seq per timestep" are the number of full Newton iterations per time step for the fully coupled Method or the number of sequential outer iterations for the SIFP and SIN methods.

Table 3.2: Nonlinear results for 50 cell one-dimensional radial model for single-phase injection (FC: Fully Coupled)

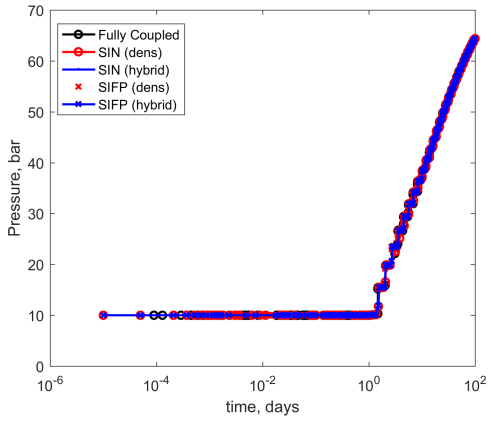
	SIFP		**SIN**		
	Pressure	**Hybrid**	**Pressure**	**Hybrid**	**FC**
Number of timesteps	16	16	16	16	16
Full Newton Iterations	-	-	-	-	36
Inner Newton Iterations	282	282	87	87	-
GMRES Iterations	-	-	319	319	-
Sequential Outer Iterations	109	109	35	35	-
Wasted Timesteps	0	0	0	0	0
Wasted Full Newtons	-	-	-	-	0
Wasted Inner Newtons	0	0	0	0	-
Newton/Seq per timestep	6.8	6.8	2.2	2.2	2.3

For the single-phase model, we omitted the results of the fixed density approach

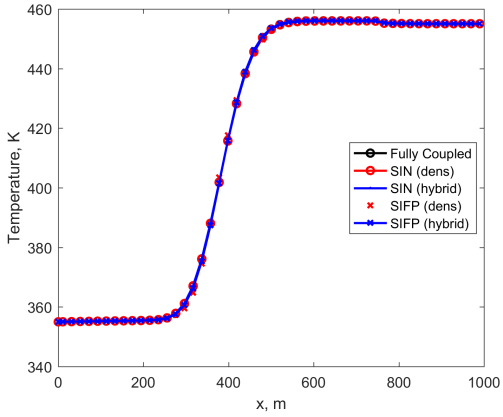
for SIFP and SIN as this was divergent. The fixed pressure and hybrid approach have identical performance for both the SIFP and SIN methods. The SIN method results in about three times fewer sequential iterations than SIFP. We also notice that the SIN method has about the same number of sequential iterations to the Newton iterations of the fully coupled method.



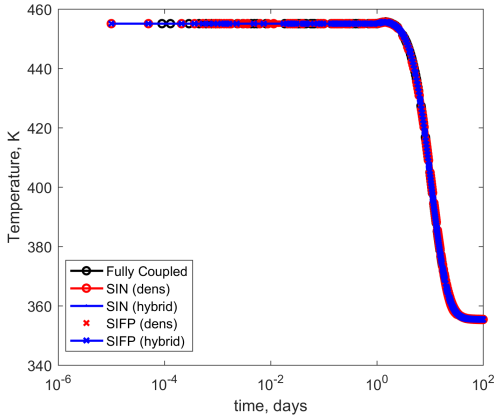
(a) Pressure profile at final timestep



(b) Pressure over time at $x = 167$



(c) Temperature profile at final time step



(d) Temperature over time at x = 167

Figure 3.5: The pressure and temperature profiles for 50 cell two-phase injection

The main difference seen in this test case in comparison to the previous test case is that this model results in two-phase behavior. Due to the phase transitions, we

see that this results in a nonmonotonic pressure solution over time at the position $x = 167$m (Figure 3.5b). The nonlinear behavior for this type of problem is described fully in Chapter 4. We see that the solutions for the different methods are consistent up to the time stepping schemes for which they are convergent for. The results for the fixed pressure scheme were omitted as it is divergent for two-phase problems. The small differences between the approaches this scheme are attributed to the different convergent time-stepping schemes.

Table 3.3: Nonlinear results for two-phase 50 cell one-dimensional radial two-phase model (FC: Fully Coupled)

	SIFP		**SIN**		
	Density	**Hybrid**	**Density**	**Hybrid**	**FC**
Number of timesteps	1803	442	234	243	304
Full Newton Iterations	28544	8349	2035	1750	1159
Inner Newton Iterations	28544	8349	2035	1750	1159
GMRES Iterations	-	-	20382	8730	-
Sequential Iterations	49374	8571	811	832	-
Wasted Timesteps	1794	432	223	232	293
Wasted Full Newtons	-	-	-	-	686
Wasted Inner Newtons	76597	14039	2281	2670	-
Newton/Seq per timestep	27.4	19.4	3.5	3.4	3.8

For the 50 cell two-phase model, we see that the SIN method outperforms the SIFP for both the fixed density and hybrid approaches. Again, we see that the fixed density SIFP has very poor nonlinear convergence, requiring about 4.1-7.7 times more timesteps than the other methods. Comparing SIFP and SIN, the total number of sequential iterations has decreased by a factor of 10 for the hybrid and a factor of 61 for the fixed density approach. It is interesting to note that although the focus

of this study was to show how the SIN outperforms the SIFP method, we see that the fully coupled approach struggles with this problem with about 1.3 times more time steps than SIN method for both coupling strategies. The issues that the fully coupled method faces for this problem are described in the Chapter 4. The wasted time steps for the fully coupled method and SIN were due to a unphysical updates to the solution. The wasted timesteps for the SIFP method were a combination of both unphysical updates and reaching the maximum number of sequential iterations.

To further investigate the wasted full and inner Newton iterations, a plot of the number of Newton iterations computed for each time step is shown (Figure 3.6). The full Newtons for the fully coupled method and the inner Newtons for SIFP and SIN are not comparable, as the full Newtons are solving for two times more residual equations with the two physics. This makes each Newton step for the fully method considerably more expensive than a single inner Newton step. However, it is useful to see qualitatively how the fully coupled method and SIN methods compare. One clear trend is that the SIN requires fewer Newton iterations than the SIFP method and at the later time, we see that the SIFP is consistently wasting about 50 inner Newton iterations. This is due to the maximum number of sequential iterations being reached by the SIFP method. This issue is not faced by SIN due to the faster outer loop sequential convergence.

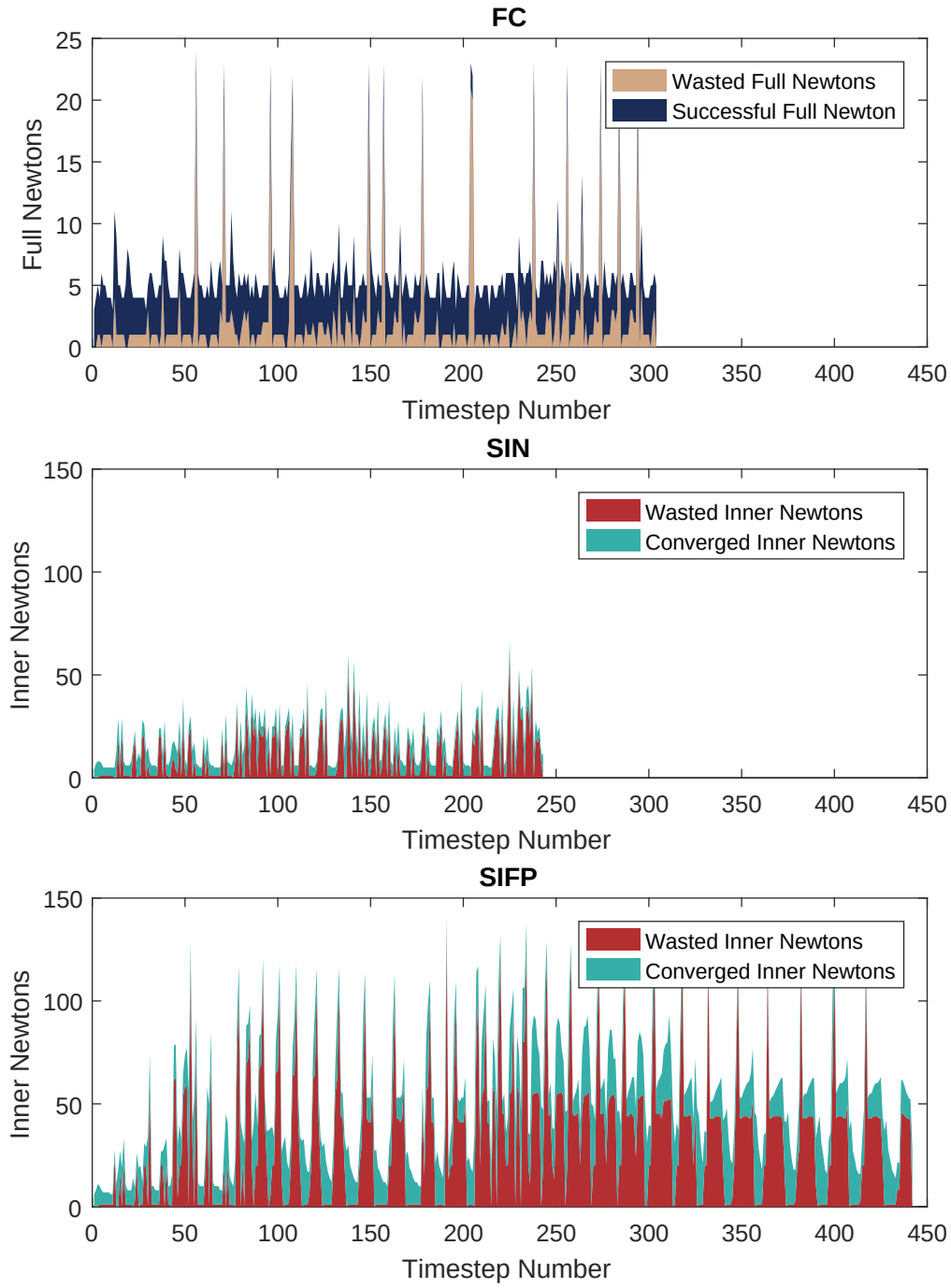


Figure 3.6: Plot of Newton iterations for each time step for the different schemes for the two-dimensional two-phase problem, here we compare the SIN, SIFP hybrid cases

From these two different one-dimensional problems, we see qualitatively a few convincing trends comparing the different approaches. Across the two flow regimes, the SIN method consistently reduced the number of sequential iterations in comparison to SIFP. The number of sequential iterations for SIN is now comparable and can be smaller than the fully coupled method. This shows qualitatively the quadratic convergence rate of the SIN outer loop updates. We also notice that when comparing different constraints for SIN, we see the same qualitative trend as SIFP, that the hybrid approach is superior to the fixed density and fixed pressure as the fixed density fails for the single-phase case and the fixed pressure fails for the two-phase case. This underscores the statement mentioned in Liu and Keyes [37] that the partition of the physical variables (and constraints) is the most interesting part of the implementation.

Two-dimensional Heterogeneous Example

The permeability and porosity distributions for this example were taken from a section of the top layer of the SPE10 model, the permeability and porosity can be seen in Figure 3.7. There are $10 \times 10 \times 1$ grid cells, each with size $20 \times 10 \times 70$m. The purpose of this test was to investigate how the different sequential strategies perform with a heterogeneous permeability field for single- and two-phase flow regimes. A constant pressure and temperature condition were specified for the cell on the top left and bottom right cells. The top left cell was set at 90 bar and 350 K. The bottom right cell was set at 60 bar and 350 K for single-phase conditions and 10 bar, 0.2 water saturation, 350 K at two-phase conditions. Both flow problems were simulated for 100 days. Figures 3.8a and 3.8b show the solutions for the single-phase and two-phase solutions at two simulation times. The relative difference between the SIN and SIFP solutions can be seen in Figure 3.9. For this study we set the maximum number of sequential iterations to be 30 and $\epsilon_F = \epsilon_T = 10^{-4}$. Similar to all the other numerical

cases, SuperLU [36] was used as the linear solver for all the linear equations solved. The tolerance for the GMRES algorithm when solving 3.38 was set to 10^{-8}.

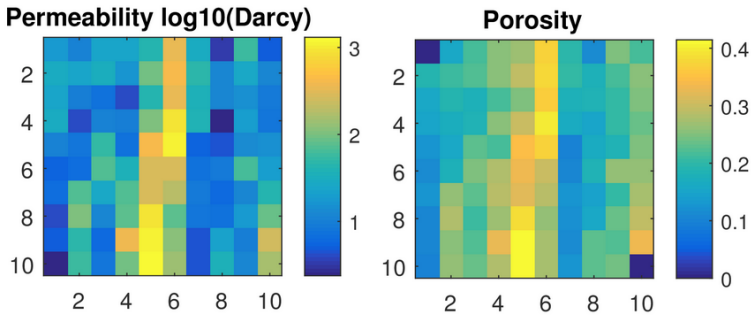
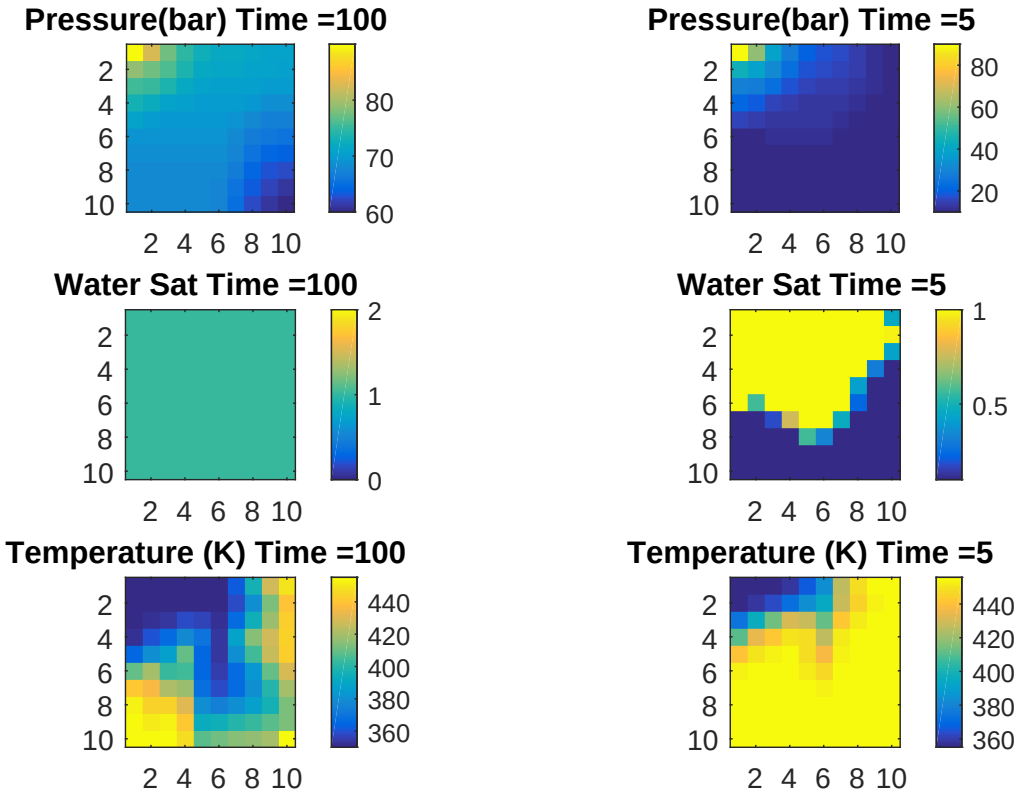
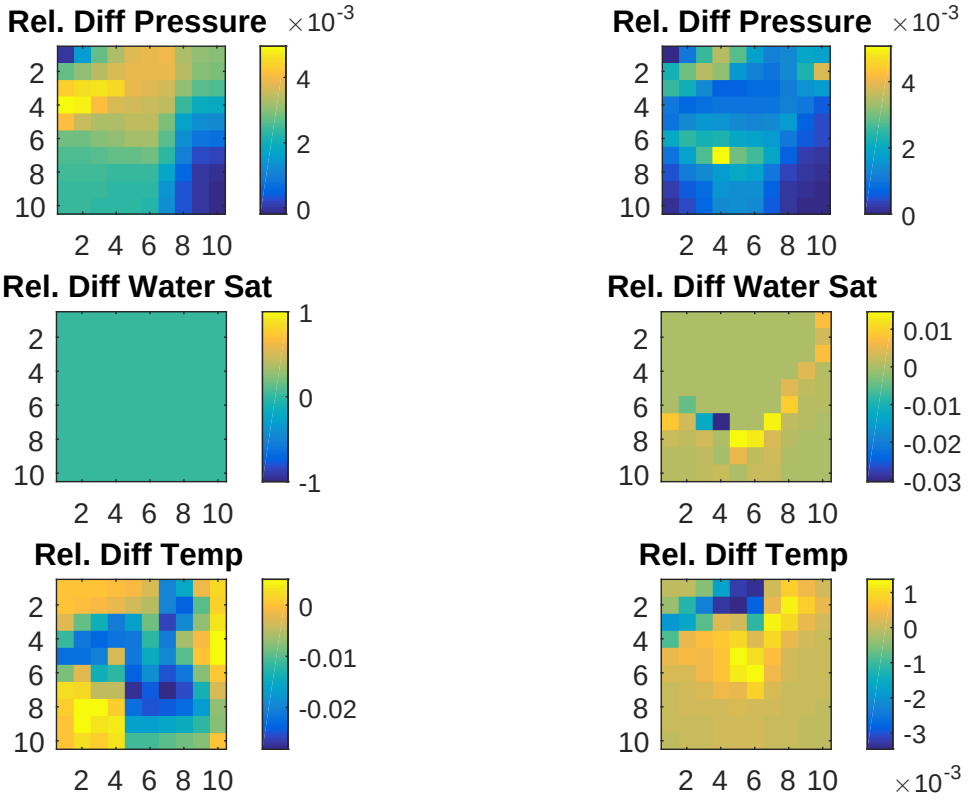


Figure 3.7: Input Permeability (Left) and Porosity (Right), Dark blue cells in porosity indicate the boundary cells



(a) Two-dimensional single-phase (b) Two-dimensional two-phase

Figure 3.8: Solution for the heterogeneous single- and two-phase models



(a) Two-dimensional single-phase

(b) Two-dimensional two-phase

Figure 3.9: Relative difference of SIFP and SIN solutions $\left(\frac{x_{SIN}-x_{SIFP}}{x_{SIFP}}\right)$ for the heterogeneous single- and two-phase models

In the one-dimensional problem, we demonstrated how the hybrid approach was superior to the other constraints for all flow regimes. Thus, we only investigated the hybrid approach for SIFP and SIN. Table 3.4 shows the nonlinear performance of the fully coupled method, hybrid SIFP, and the hybrid SIN method. For the single-phase problem, we see that the number of sequential iterations for SIFP is 7.2 times more than SIN. We also notice that there were six wasted iterations. However, for SIN, we see that the number of sequential outer loop iterations (36) is identical to the fully coupled Newton iterations.

For the two-phase problem, where cold water is invading a two-phase reservoir, we again see the effectiveness of SIN. We see that SIFP requires 27 times more outer loop sequential iterations than the SIN method. Similar to the one-dimensional problem, because the fully coupled approach struggles with this type of physics problem, we see that SIN has better nonlinear convergence than the fully coupled method and requires fewer timesteps. Again, we see a similar number of sequential iterations per timestep for SIN and the fully coupled method. This is larger than the other problems examined as for this particular flow regime, there is a stronger nonlinearity from the phase change and two-phase flow.

Table 3.4: Nonlinear results for heterogeneous two-dimensional model (FC: Fully Coupled)

	Single-phase			**Two-phase**		
	FC	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**
Number of Timesteps	20	25	20	129	373	96
Full Newton Iterations	36	-	-	607	-	-
Inner Newton Iterations	-	295	84	-	9531	1089
GMRES Iterations	-	-	325	-	-	6565
Sequential Iterations	-	259	36	-	10421	381
Wasted Timesteps	0	6	0	109	353	76
Wasted Full Newtons	0	-	-	555	-	-
Wasted Inner Newtons	-	282	0	-	14369	1550
Newton/Seq per timestep	1.8	10.4	1.8	4.7	27.9	4.0

To compare the Newton iterations for the three different methods for the two-phase case, we again plot the number of Newton iterations for the fully coupled, SIN and SIFP method in Figure 3.10. Here we see even more clearly that the SIFP has a large number of timesteps where 50 Newton iterations are wasted. This is again due

to the maximum number of sequential iterations being reached. For the SIN, we do not see that same consistent trend of time steps wasted, as the time steps wasted for the SIN method is because of unphysical updates. We see that because SIN is able to overcome the outer loop convergence issues faced by SIFP, it is able to converge to a timestepping scheme that requires less timesteps than the fully coupled method. From this Figure 3.10 we see clearly how the poor sequential convergence of the SIFP method can affect the time steps of the problem.

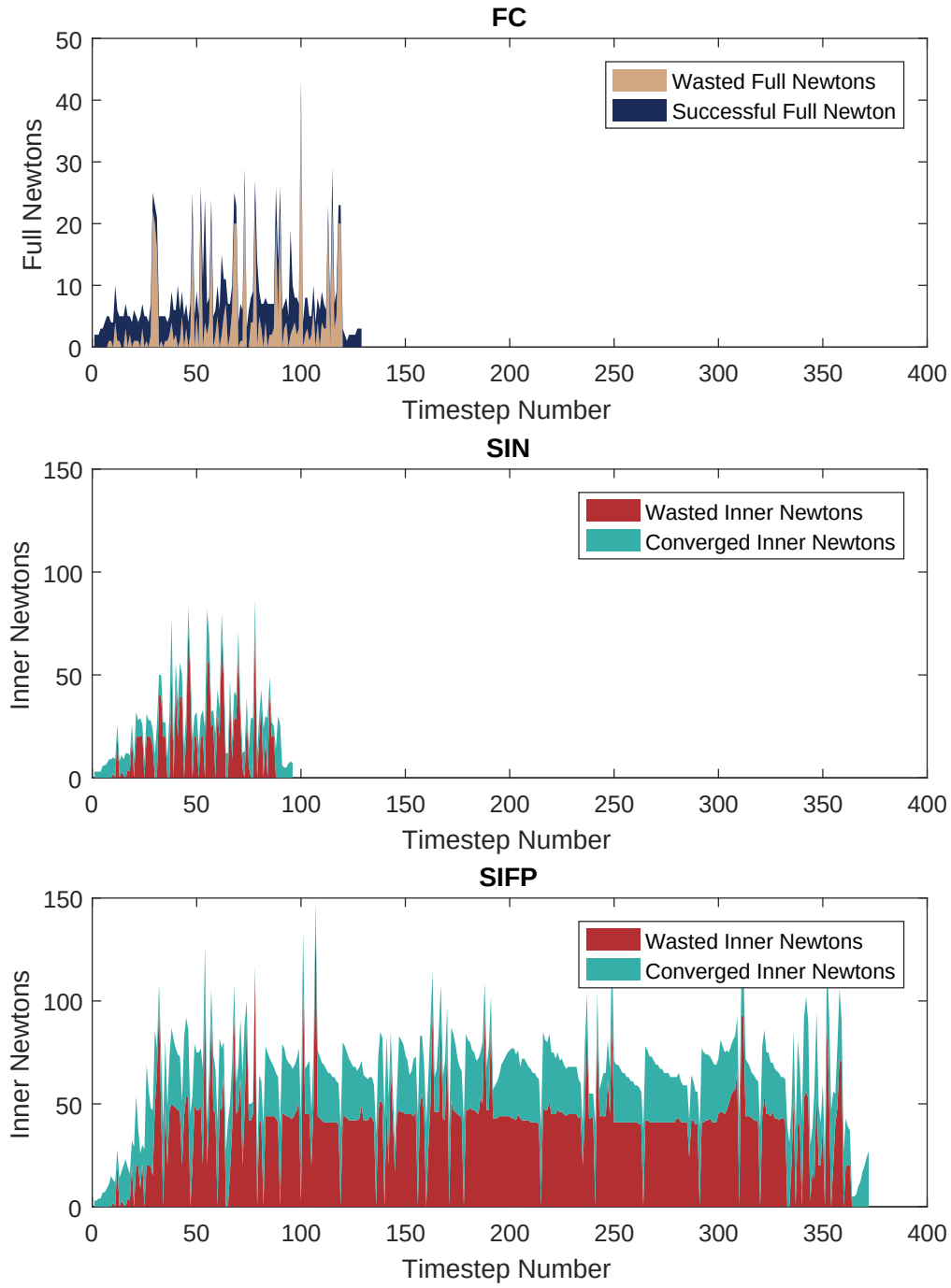


Figure 3.10: Plot of Newton iterations for each time step for the different schemes for the two-dimensional two-phase problem

Increasing the complexity of the problem with heterogeneity and two-dimensional flow further improves the speed up of SIN compared to SIFP. As shown similarly in Chapter 2, the SIFP with a hybrid approach has been shown to struggle as more complexities such as heterogeneity are added. However, for SIN, it is able to have a comparable number of sequential iterations with the Newton iterations for a fully coupled method. This is due to the quadratic convergence rate of which Newton's method is able to take advantage. Here the six wasted time steps for the SIFP single-phase problem were due to the maximum number of sequential iterations reached. For the fully coupled method and SIN two-phase cases, the wasted time steps were due to unphysical updates in the inner loop solutions.

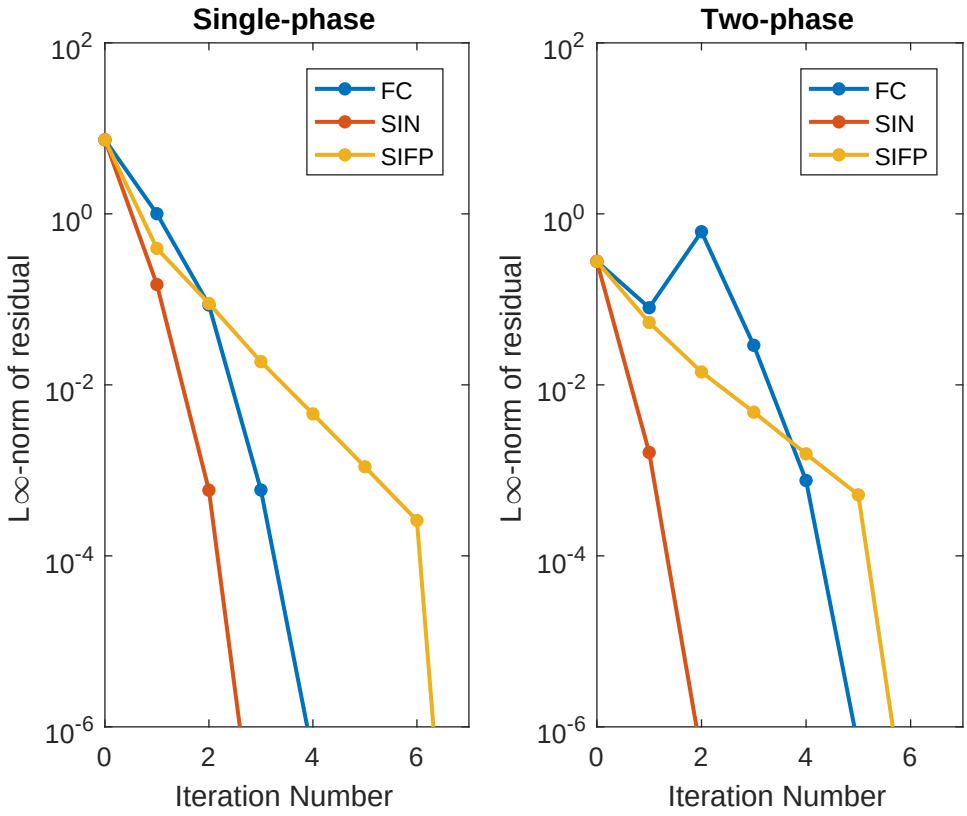


Figure 3.11: Plot of L_∞-norm of the flow and thermal residual after each sequential (SIFP/SIN) or Newton (FC: Fully Coupled) iteration

Figure 3.11 shows a plot of the L_∞-norm of the residual after each sequential or Newton iteration. These residuals were taken at $T = 10^{-2}$ days. We see that for both the single- and two-phase problems, we have the SIN method converging faster than SIFP. For the single-phase problem, we see that the fully coupled and SIN method have similar convergence profiles, with the SIN method converging in one less iteration. For the two-phase problem we notice that the fully coupled method took only one less iteration that SIFP, this is due to the difficulty for the fully coupled method for this type of cold water injection problem. This nonlinear convergence issue is fully described in Chapter 4.

3.5 Flow-Mechanics Problem

3.5.1 Governing Equations

In this section, we consider the interaction between the flow of a single component, namely water, and the mechanics of the rock that surrounds it. Assuming constant temperature, the water component here can only exist in a single-phase (liquid water). The flow equation takes the form:

$$\frac{\partial}{\partial t}\left(\phi\rho_w\right) - \nabla.\left(\rho_w \boldsymbol{v}_w\right) - Q_M = 0 \tag{3.44}$$

where:

- ϕ is the porosity of the rock;
- ρ_l is the mass density of the liquid water;
- $\boldsymbol{v}_w$ is the velocity of the water;
- Q_M is the mass source/sink term.

The density ρ_w of the liquid water is a function of pressure. To model the flow rate of each phase, Darcy's law is used to describe the flow through the porous medium (3.25). The porosity ϕ depends on both the pressure and the deformation of the rock, which is described by the mechanics equation below. The quasistatic momentum conservation equation for the aggregate volume (rock skeleton and fluid) is written as:

$$\nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} = 0 \tag{3.45}$$

where $\rho = \rho_s(1-\phi) + \rho_f \phi$ is the overall mass density, ρ_s is the density of the rock skeleton and ρ_f is the cell-averaged fluid density. The total stress tensor $\boldsymbol{\sigma}$ consists of the both the fluid and rock-skeleton stresses [11].

$$\boldsymbol{\sigma} = \mathbb{C}\epsilon^e - \boldsymbol{b}p \tag{3.46}$$

where ϵ^e is the second-order elasticity strain tensor, $\mathbb{C}$ is the fourth-order tensor elasticity moduli tensor, $\boldsymbol{b} = \mathbf{1}b$ is the second-order tensor of Biot coefficients b and p is the fluid pressure. From 'small deformation' theory and assuming that the total strain tensor is only composed of elastic contribution, ignoring plastic and thermal effects:

$$\epsilon = \epsilon^e \tag{3.47}$$

We can rearrange from Equation (3.46), (3.47):

$$\boldsymbol{\sigma} = \mathbb{C}\epsilon - \boldsymbol{b}p \tag{3.48}$$

The total strain is defined as:

$$\epsilon = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u}\right) \tag{3.49}$$

The mechanical effect on the flow equation is captured through the porosity's relationship with stress and strain. We follow Coussy [11] to capture the porosity change through the ratio of the volume of connected porous space to the total volume:

$$\phi = \phi_0 + \frac{(b-\phi_0)(1-b)}{K_d}(P - P_0) + b(\epsilon_v - \epsilon_{v,0}) \tag{3.50}$$

where K_d is the local drained bulk modulus, $\epsilon_v = \mathrm{tr}(\epsilon)$ is the volumetric total strain, ϕ_0, P_0, $\epsilon_{v,0}$ are the reference porosity, pressure and volumetric strain. Note that the flow equation (3.44) is coupled to (3.45) through the porosity ϕ, and the momentum balance equation (3.45) depends on (3.44) through the fluid pressure p.

3.5.2 Fully Coupled Formulation

For the fully coupled solution, we discretize the flow equations (3.44) fully implicitly in time (backward Euler) and using the finite volume method in space [3], and the momentum balance equation (3.45) using P^1 finite elements in space [86]. Choosing the pressure and displacement as primary variables and eliminating all other quantities using Equations (3.23), (3.25), (3.46)–(3.50), we arrive at the discrete algebraic problem at time step $n+1$, which takes the form

$$\begin{cases} \tilde{R}_F(p_{n+1}, \boldsymbol{u}_{n+1}) := R_F(p_{n+1}, \sigma(p_{n+1}, \boldsymbol{u}_{n+1})) = 0, \\ R_u(p_{n+1}, \boldsymbol{u}_{n+1}) = 0, \end{cases} \tag{3.51}$$

where:

- $p_{n+1} \in \mathbb{R}^{N_c}$ is the vector of pressures at each of the N_c cell centers at time t_{n+1};
- $\boldsymbol{u}_{n+1} \in \mathbb{R}^{N_D N_v}$ is the displacement vector at each of the N_v vertices in the N_D dimensions;

- $R_F : \mathbb{R}^{N_c} \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is the residual form of the mass conservation equation (3.44) as a function of the mean stress σ; [1]
- $\sigma : \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_V} \to \mathbb{R}^{N_c}$ is a function that computes the cell center mean stress based on the pressure and the displacement field;
- $R_u : \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_v} \to \mathbb{R}^{N_D N_v}$ is the residual form of the momentum balance for the mechanics equations (3.45).

Then the fully coupled algorithm consists of applying Newton's method to the system (3.51), using the solution of the previous time step $(p_n, \boldsymbol{u}_n)$ as the initial guess. More precisely, the iterative process is as follows:

1. Solve for $p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1}$ using:

$$\begin{bmatrix} \frac{\partial \tilde{R}_F}{\partial p} & \frac{\partial \tilde{R}_F}{\partial \boldsymbol{u}} \\ \frac{\partial R_u}{\partial p} & \frac{\partial R_u}{\partial \boldsymbol{u}} \end{bmatrix}_{n+1}^{k} \begin{bmatrix} \Delta p^k \\ \Delta \boldsymbol{u}^k \end{bmatrix} = - \begin{bmatrix} \tilde{R}_F(p_{n+1}^k, \sigma(p_{n+1}^k, \boldsymbol{u}_{n+1}^k)) \\ R_u(p_{n+1}^k, \boldsymbol{u}_{n+1}^k) \end{bmatrix} \tag{3.52}$$

   where $\Delta p^k = p_{n+1}^{k+1} - p_{n+1}^k$, $\Delta \boldsymbol{u}^k = \boldsymbol{u}_{n+1}^{k+1} - \boldsymbol{u}_{n+1}^k$, and the Jacobian matrices $\frac{\partial \tilde{R}_F}{\partial p}, \frac{\partial \tilde{R}_F}{\partial \boldsymbol{u}}, \frac{\partial R_u}{\partial p}, \frac{\partial R_u}{\partial \boldsymbol{u}}$ are all evaluated at $(p_{n+1}^k, \boldsymbol{u}_{n+1}^k)$.

2. Step 1 is repeated until convergence:

$$\left|\left|\tilde{R}_F(p_{n+1}^{k+1}, \sigma(p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1}))\right|\right|_\infty \leq \epsilon_p \quad \text{and} \quad \left|\left|R_u(p_{n+1}^{k+1}, \boldsymbol{u}_{n+1}^{k+1})\right|\right|_\infty \leq \epsilon_u. \tag{3.53}$$

3.5.3 Fixed Stress Sequential-implicit Formulation

In this section, we demonstrate the effectiveness of the SIN method for the fixed stress formulation for flow and mechanics. Here we combine the finite-volume method [3]

[1] This is different from the R_F for the flow-thermal problem. However, since we will not talk about both problems simultaneously, no confusion should arise, so we prefer to use the same notation for both flow equations.

to discretize the flow equation (Equation 3.44) and the finite-element method for the momentum balance (Equation 3.45) [86]. We follow the same framework for the sequential-implicit method as described in [18] for isothermal flow and mechanics problems. Here we utilize the geomechanics implementation within AD-GPRS [19, 20].

We use the solution of the previous timestep as the initial guess $x^0_{n+1} = (p^0_{n+1}, \boldsymbol{u}^0) = (p_n, \boldsymbol{u}_n) = x_n$. Where $p^0 \in \mathbb{R}^{N_c}$ is the pressure at the cell centers and $\boldsymbol{u}^0 \in \mathbb{R}^{N_D N_v}$ is the displacement vector at each of the N_v vertices in the N_D dimensions. For simplicity, we will now drop the subscript $n+1$ and all following terms represent the solution at the $n+1$ timestep.

The fixed-stress sequential iterative process is as follows:

1. Compute $\sigma^k = \sigma(p^k, \boldsymbol{u}^k)$
2. Solve for p^* where $R_F(p^*, \sigma^k) = 0$, where convergence is defined as: $\left|\left|R_F(p^*, \sigma^k)\right|\right|_\infty \leq \epsilon_p$
3. Solve for $\boldsymbol{u}^*$ where $R_u(p^*, \boldsymbol{u}^*) = 0$ where convergence is defined as: $||R_u(p^*, \boldsymbol{u}^*)||_\infty \leq \epsilon_u$
4. Update x^{k+1} by fixed-point iteration or Newton's method
5. Repeat steps 1-4 until converged:

$$\left|\left|R_F(p^{k+1}, \sigma(p^{k+1}, \boldsymbol{u}^{k+1}))\right|\right|_\infty \leq \epsilon_p \text{ and } \left|\left|R_u(p^{k+1}, \boldsymbol{u}^{k+1})\right|\right|_\infty \leq \epsilon_u \tag{3.54}$$

3.5.4 Sequential-implicit Fixed-point Fixed Stress Algorithm

The update for the SIFP fixed stress algorithm is simply using the solutions from the flow and mechanics residual equations as the update:

$$x^{k+1} = (p^*, \boldsymbol{u}^*) \tag{3.55}$$

3.5.5 Sequential-implicit Newton Fixed Stress Algorithm

To obtain the SIN update, the below nonlinear system is solved using Newton's method:

$$\mathcal{F}(p^k, \boldsymbol{u}^k) = \begin{bmatrix} p^k - p^*(p^k, \boldsymbol{u}^k) \\ \boldsymbol{u}^k - \boldsymbol{u}^*(p^k, \boldsymbol{u}^k) \end{bmatrix} \tag{3.56}$$

To solve this nonlinear system using Newton's method we need to compute the matrix-vector product for the Jacobian $\frac{\partial \mathcal{F}}{\partial x}$ and an arbitrary vector $v = (p, \boldsymbol{u})$, $p \in \mathbb{R}^{N_c}$, $\boldsymbol{u} \in \mathbb{R}^{N_D N_v}$, $v \in \mathbb{R}^{N_c} \times \mathbb{R}^{N_D N_v}$:

$$\frac{\partial \mathcal{F}}{\partial x} v = \begin{bmatrix} p - w_1 \\ \boldsymbol{u} - w_2 \end{bmatrix} \tag{3.57}$$

The steps to compute to obtain the update Δx^k given a $x^k = (p^k, \boldsymbol{u}^k)$ are as follows:

1. Compute $\sigma_k = \sigma(p^k, \boldsymbol{u}^k) \in \mathbb{R}^{N_c}$
2. Solve for p^*, from $R_F(p^*, \sigma^k) = 0$, at convergence ($\left|\left|R_F(p^*, \sigma(p^*, \boldsymbol{u}^k))\right|\right|_\infty \leq \epsilon_F$) store:
   - $J_{11} = \frac{\partial R_F}{\partial p}\Big|_{\partial \sigma = 0} (p^*, \sigma^k) \in \mathbb{R}^{N_c \times N_c}$ or store the LU factors for multiplying J_{11}^{-1}, that was used to solve $R_F(p^*, \sigma^k) = 0$.
   - $J_{12} = \frac{\partial R_F}{\partial \sigma}\big|_{\partial p = 0} (p^*, \sigma^k) \in \mathbb{R}^{N_c \times N_c}$
   - $J_\sigma = \frac{\partial \sigma}{\partial x}(p^*, \sigma^k) \in \mathbb{R}^{N_c \times (N_c + N_D N_v)}$

3. Solve for $\boldsymbol{u}^*$, from $R_u(p^*, \boldsymbol{u}^*) = 0$, at convergence ($||R_u(p^*, \boldsymbol{u}^*)||_\infty \leq \epsilon_u$) store:

   - $J_{21} = \frac{\partial R_u}{\partial p}\Big|_{\partial u=0}(p^*, \boldsymbol{u}^*) \in \mathbb{R}^{N_D N_v \times N_c}$
   - $J_{22} = \frac{\partial R_u}{\partial u}\Big|_{\partial p=0}(p^*, \boldsymbol{u}^*) \in \mathbb{R}^{N_D N_v \times N_D N_v}$ or store the LU factors for multiplying J_{22}^{-1}, that was used to solve $R_u(p^*, \boldsymbol{u}^*)$

4. Solve the system of equations:

$$\frac{\partial \mathcal{F}}{\partial x}\Delta x^k = -\mathcal{F} \tag{3.58}$$

   using GMRES and the matrix-vector product defined as:

$$\frac{\partial \mathcal{F}}{\partial x}v = \begin{bmatrix} p - w_1 \\ \boldsymbol{u} - w_2 \end{bmatrix} \tag{3.59}$$

   Where:

   - $w_1 = -(J_{11})^{-1}(J_{12}J_\sigma v) \in \mathbb{R}^{N_c}$
   - $w_2 = -(J_{22})^{-1}J_{21}w_1 \in \mathbb{R}^{N_D N_v}$ Here if steps 2 and 3 are solved with a direct solver, we can utilize the same LU factors for the multiplication of J_{11}^{-1} and J_{22}^{-1}

5. Update:

$$\begin{bmatrix} p^{k+1} \\ \boldsymbol{u}^{k+1} \end{bmatrix} = \Delta x^k + \begin{bmatrix} p^k \\ \boldsymbol{u}^k \end{bmatrix} \tag{3.60}$$

6. Repeat steps 1-5 until:

$$\left|\left|R_F(p^{k+1}, \sigma(p^{k+1}, \boldsymbol{u}^{k+1}))\right|\right| \leq \epsilon_F \text{ and } \left|\left|R_u(p^{k+1}, \boldsymbol{u}^{k+1})\right|\right| \leq \epsilon_u \tag{3.61}$$

3.5.6 Mandel's Problem

We considered Mandel's consolidation problem in two-dimensions [39, 7]. Mandel's problem is a two-dimensional problem with a homogeneous and isotropic poroelastic rock fixed with two impermeable, rigid and frictionless plates on the top and bottom boundaries. The grid was set to have 20 cells in the x and z directions.

Table 3.5: Rock and fluid properties used for Mandel's problem.

Property	**Value**	**Unit**
Reference porosity	37.5	%
Young's modulus	$\{10^9, 2 \times 10^8, 10^8\}$	Pa
Biot's constant	1.0	-
Poisson ratio	0.25	-
Undrained Poisson ratio	0.47	-
Permeability,	1	md
Fluid viscosity	9.81×10^{-5}	$\text{Pa} \cdot \text{s}$
Fluid compressibility	4.4×10^{-10}	Pa^{-1}
Reference fluid density	1000	kg/m^3

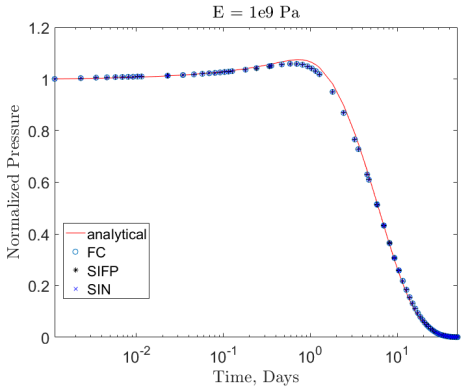
For the SIFP method it has been shown that the coupling strength of the problem [28, 30, 29] is related to the parameter:

$$\tau \equiv \frac{b^2 M}{K_{dr}} \tag{3.62}$$

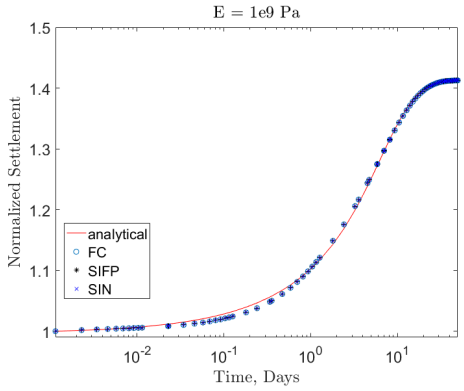
where b is Biot's coefficient, M is the Biot modulus and K_{dr} is related to the bulk modulus. We can increase the coupling strength by decreasing the Young's modulus of the problem [18]. Here we examined the results for three different Young's moduli, $10^9, 2 \times 10^8, 10^8$ Pa. Here a lower Young's modulus would result in a greater coupling strength, thus for SIFP we would expect this to require more outer loop iterations to

converge. In all cases, the convergence tolerance was set to $\epsilon_F = \epsilon_u = 10^{-6}$ and the maximum number of sequential iterations was set to 30. The tolerance for the SIN GMRES solution step was set to 10^{-8}. The linear solver for multiplying by J_{11}^{-1} and J_{22}^{-1} at all steps was SuperLU [36], this was used to decouple any effects the linear solver would have on the sequential updates.

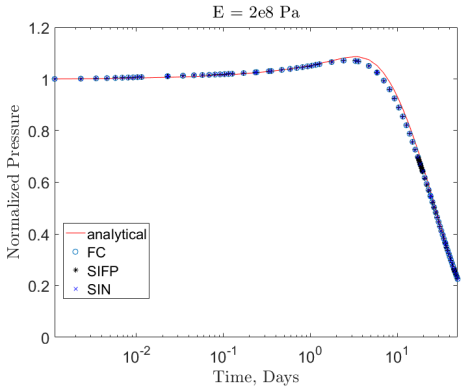
As shown in Figure 3.12, the fully coupled, SIFP and SIN solutions all match well with the analytical solution. Mandel's problem is very close to a linear problem, with most of the nonlinearity coming from the very small compressibility of the fluid. The full nonlinear results for Mandel's problem is shown in Table 3.6.



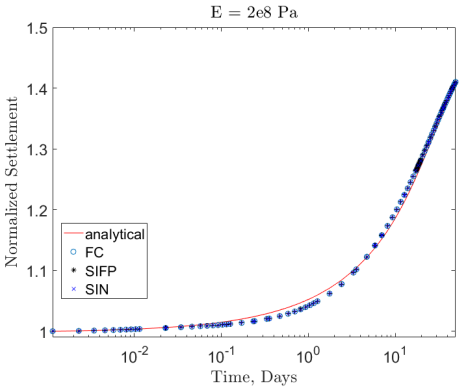
(a) Pressure profile at x = 0 m,y = 100 m ($E = 1 \times 10^9$ Pa)



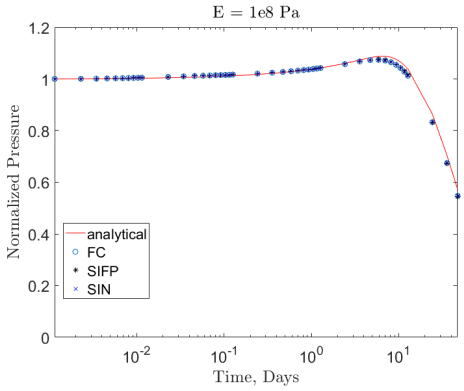
(b) Displacement profile at x = 0 m,y = 100 m ($E = 1 \times 10^9$ Pa)



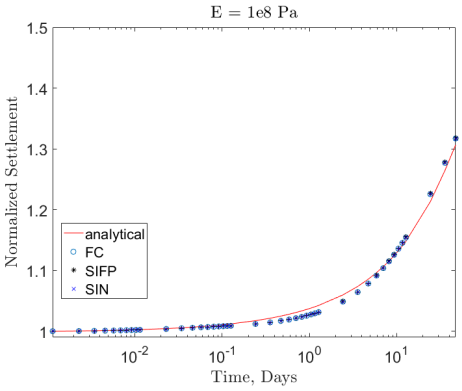
(c) Pressure profile at x = 0 m,y = 100 m ($E = 2 \times 10^8$ Pa)



(d) Displacement profile at x=0 m,y = 100 m ($E = 2 \times 10^8$ Pa)



(e) Pressure profile at x = 0 m,y = 100 m ($E = 1 \times 10^8$ Pa)



(f) Displacement profile at x = 0 m,y = 100 m ($E = 1 \times 10^8$ Pa)

Figure 3.12: Plot of pressure and displacement profiles for 20×20 grid for Mandel's problem

Table 3.6: Nonlinear results for Mandel's problem (FC: Fully Coupled, SIFP: Sequential-implicit fixed-point , SIN: Sequential-implicit Newton)

	$E = 10^9$ Pa			$E = 2 \times 10^8$ Pa			$E = 10^8$ Pa		
	FC	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**	**FC**	**SIFP**	**SIN**
Number of Timesteps	90	90	90	92	154	92	93	860	93
Full Newton Iterations	102	-	-	102	-	-	108	-	-
Inner Newton Iterations	-	766	418	-	2519	516	-	10746	549
Sequential Outer Iterations	-	384	226	-	3339	291	-	29569	313
GMRES Iterations	-	-	1958	-	-	3585	-	-	4484
Wasted Timesteps	0	0	0	0	65	0	0	791	0
Wasted Full Newtons	0	-	-	0	-	-	0	-	-
Wasted Inner Newtons	-	0	0	-	2795	0	-	34046	0
Newton/Seq per timestep	1.1	4.3	2.5	1.1	21.7	3.2	1.1	34.3	3.4

Here we see even that for an almost linear problem, the SIFP method can struggle heavily due to the strong coupling between the two subproblems. This shows that the SIN method is able to overcome these issues the SIFP faces. Although the fully coupled method outperforms both SIN and SIFP methods, this still demonstrates the focus of this study that the SIN method is able to overcome the difficulties faced by the SIFP method. As expected, as we decrease the Young's modulus, the number of sequential iterations for SIFP increases and is convergent only for a very small time step. For the SIN approach, we see a milder trend, where although there is a slight increase in sequential iterations with coupling strength, due to the quadratic convergence of Newton's method, this is still at about 2-3 outer sequential iterations per timestep. We see that with SIN, we can get nearly two orders of magnitude less sequential iterations than the SIFP method when the problem is strongly coupled ($E = 10^8$). We see that this slow convergence issue is further intensified by the number of wasted timesteps for the nonconvergent timesteps. These nonconverging timesteps were due to the maximum number of outer sequential iterations (30) being

reached. Figure 3.13 shows the residual error for sequential iteration for the first time step ($\Delta t = 0.001$ days). We see that as we increase the coupling strength of the problem (increase Young's modulus), as expected the outer loop iterations for the fixed-point iteration increases. However, we see that for both SIN and the fully coupled method, the convergence profile remains relatively similar.

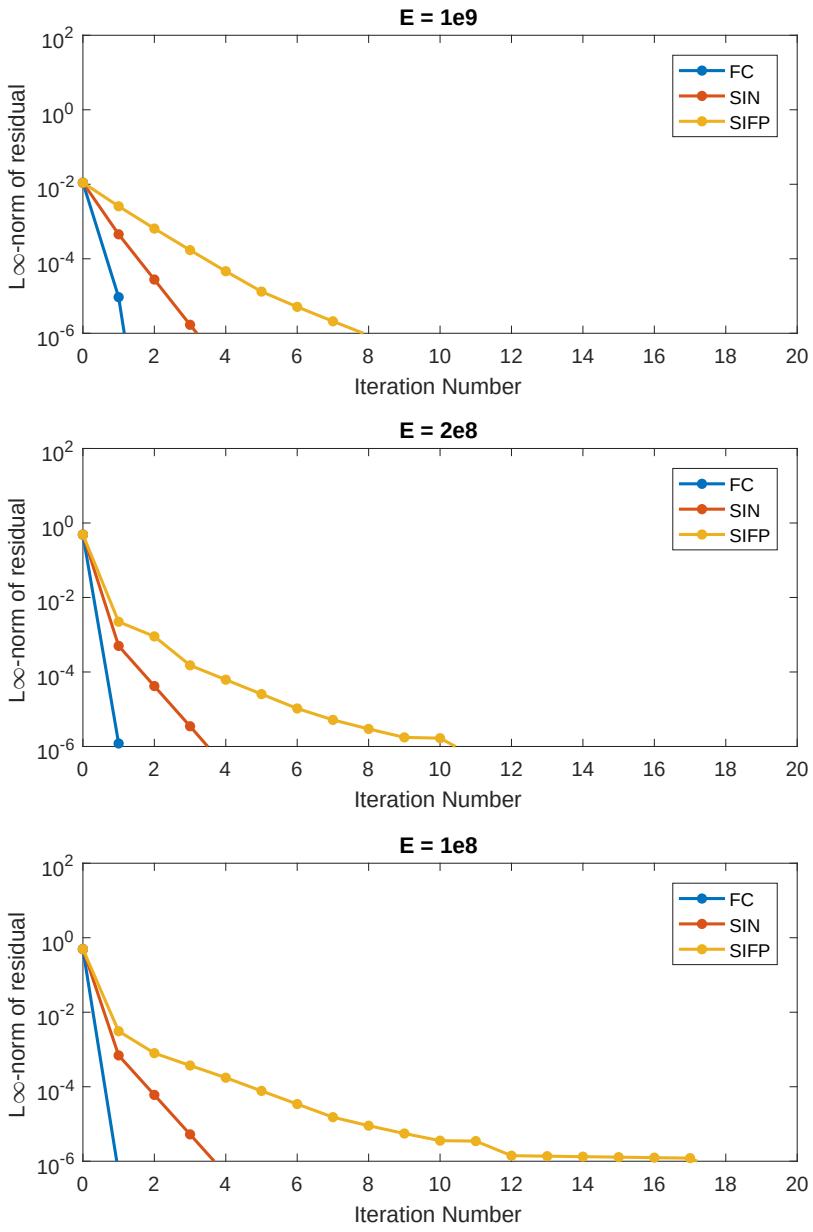


Figure 3.13: Plot of the L_∞-norm for the flow and mechanics residual after each sequential (SIFP, SIN) or Newton (Fully Coupled) iteration for different Young's modulus

3.6 Chapter Summary

Sequential-implicit simulations currently employ a fixed-point iteration. This fixed-point approach has only a linear convergence rate and can be problematic for multiphysics problems with strong coupling between the subproblems that are split. A sequential-implicit Newton's method was presented and was shown to improve this convergence rate to a quadratic convergence rate.

This sequential-implicit Newton method was demonstrated on two different multiphysics problems: a flow-thermal and a flow-mechanics problem. For all the test cases investigated, the sequential-implicit Newton outperformed the sequential-implicit fixed-point iteration approach in terms of the sequential outer loop convergence. For problems that have more complexities or a stronger coupling between the subproblems, the sequential-implicit Newton can have an order of magnitude decrease in the number of sequential iterations. The sequential-implicit Newton convergence rate is also shown to be qualitatively similar to a fully coupled approach where Newton's method is applied to the entire multiphysics problem. The generality of this method shown through the effectiveness of the flow-mechanics problem shows that the sequential-implicit Newton's method could be effective for any multiphysics problems.

These advances in the design of sequential-implicit methods for geothermal simulation are important to the efficiency and understanding of the nonlinear issues in geothermal simulation. Now with a suitable sequential-implicit method to split the flow and thermal physics in geothermal simulations, specialized solvers could be designed to better improve the solution of the individual physical subproblem.

Chapter 4

Sequential-implicit Nonlinear Solver for the Condensation Problem

4.1 Introduction

In Chapters 2 and 3, we described different sequential-implicit methods for splitting the flow and thermal subproblems in the geothermal problem. Those investigations were targeted at designing a sequential-implicit scheme that was stable and convergent for all geothermal reservoir models. In this chapter, we instead focus on a specific but widespread nonlinear issue in geothermal simulations. Here we present a sequential-implicit strategy that can provide a better initial guess for the fully coupled method when simulating the condensation problem in geothermal simulation. An improvement in the convergence for the condensation problem would improve the robustness and convergence of geothermal reservoir simulations. Consequently, this would improve the capability of geothermal reservoir simulators to forecast and be

used as management tools for geothermal fields.

The focus of the work described in this chapter was to improve the convergence and to provide analysis and insight into this condensation problem. This condensation problem is a result of a "negative compressibility" issue. This "negative compressibility" behavior occurs in cells that contain two fluid phases: water (liquid) and steam (vapor). This problem is most pronounced in pure-water systems, whereby depending on the pressure and temperature conditions, the single component (H_2O) may exist as a single-phase liquid, a single-phase vapor, or as a two-phase (liquid-vapor) system. Coats [10] described this phenomenon using a simple single-cell problem, where cold water is injected at a fixed pressure into a cell with saturated steam. As condensation occurs, the cell pressure decreases, and that enhances the inflow of cold water. Even so, the cell pressure continues to decline as the steam condenses. As the last drops of steam condense, and liquid water, which has a small compressibility, occupies the entire pore space enclosed by the cell, the cell pressure rises dramatically. The cell pressure ultimately rises to the injection pressure, and the inflow of water ceases. This behavior can be explained as follows. During the condensation process, the reduction in the vapor-phase volume overwhelms the expansion of the fluids due to compressibility effects, and the overpressure displays this "negative compressibility" behavior.

This "negative compressibility" effect has also been discussed in more detail by Pruess [53] and Falta [14]. In those analyses, the "negative compressibility" phenomenon was attributed to the idealization that there is complete thermodynamic equilibrium combined with a computational mesh (grid) of finite size. It was found that this idealization results in spurious pressure variation in the cells that contain the two-phase front. These pressure oscillations present a severe limitation for the nonlinear convergence of these problems [21]. The solution presented in [21] was to artificially restrict any flow of cold water into the cells that are steam-saturated. For

their specific flow regime where steam is injected into cold water, Gudbjerg et al. [21] circumvented the "negative compressibility" issue by preventing the unphysical backflow of cold water. However, for the case where water is being injected into steam with a large temperature difference, their approach would produce unphysical results. This limits the application of their solution in most geothermal fields where colder water is commonly injected into a steam reservoir.

Wang [73] presented an analysis of the "negative compressibility" issue for fully implicit formulations. In that analysis, a stability criterion for the timestep was developed to ensure convergence of the fully implicit solution, thus preventing unnecessary timestep cuts. However, the derived stability criterion enforces a severe limitation on the allowable timestep size. This work investigated a nonlinear preconditioner that can be applied to the fully coupled, fully implicit solution. The preconditioner avoids the severe timestep limitation on this problem, or unphysical solutions, as presented in the other works. The preconditioner is based on a sequential fully implicit approach that provides a good initial guess to Newton's Method. This initial guess avoids the issues related to the "negative compressibility".

The rest of this chapter is organized as follows: in Section 4.2 a single-cell nonlinear analysis is shown. This section provides the foundational understanding of the "negative compressibility" problem and how the sequential-implicit scheme is able to overcome it. Section 4.3 describes the preconditioning algorithm that provides a better initial guess for the fully coupled Newton solution process. Section 4.4 describes the implementation of this nonlinear solver in AD-GPRS. Section 4.5 presents a nonlinear analysis and results for a one-dimensional case. Section 4.6 presents two-dimensional results for this approach. A chapter summary is given in Section 4.7.

4.2 Single Cell Nonlinear Analysis

The apparent "negative compressibility" problem can be demonstrated most clearly through a nonlinear analysis of a single-cell problem, as was described by Coats [10]. Here, we demonstrate how the coupling of the mass and energy conservation equations for a single cell leads to diverging nonlinear solutions for large timesteps. This single-cell problem provides the foundation for the nonlinear preconditioning method of the full physics in multidimensional problems.

Consider a single cell with a saturated mixture of steam and water. Cold water is introduced into the cell at a fixed pressure. We make the following assumptions: (1) the internal energy is equal to the absolute enthalpy of the system, (2) there is no energy contribution from the rock, and (3) the rock is incompressible. Without these simplifications, the full physics would still exhibit the same nonlinear convergence issues, but the analysis would be more difficult. The mass and energy conservation equations for the single cell are:

$$V\frac{\partial \rho_t}{\partial t} - \Upsilon(p_{\text{in}} - p) = 0 \tag{4.1}$$

$$V\frac{\partial \rho_t h}{\partial t} - H_{inj}\Upsilon(p_{\text{in}} - p) = 0 \tag{4.2}$$

$$\rho_t = S_w\rho_w + S_s\rho_s \tag{4.3}$$

where:

- Υ is the transmissibility between the cell and the source term
- p_{in} is the constant pressure of the source term
- p is the pressure of the cell

- h is the enthalpy of the cell
- t is time
- S_w, S_s is the saturation of liquid water and steam
- ρ_w, ρ_s is the water and steam saturated densities
- ρ_t is the total density of the cell
- H_{inj} is the enthalpy of the injected fluid
- V is the pore volume of the cell

As an example, we take the initial condition as:

- $p = 10$ bar, $S_s = 0.9$, $h = 87.2$ kJ/kg
- $p_{\text{in}} = 90$ bar, $H_{inj} = 34.5$ kJ/kg

4.2.1 Primary Variables

For this analysis, we consider the solution using the primary variables of pressure, p, and enthalpy, h [15]. The choice of these primary variables is preferred for this analysis, since it does not entail variable switching upon the appearance/disappearance of a fluid phase[9].

For the pressure-enthalpy formulation, the phase state is determined by comparing the enthalpy, h, with the saturated enthalpies h_w, h_s, both of which are functions of pressure.

$$\text{Phase State} := \begin{cases} \text{Single-phase liquid water} & h < h_w(p) \\ \text{Two-phase} & h_w(p) < h < h_s(p) \\ \text{Single-phase superheated steam} & h > h_s(p) \end{cases} \tag{4.4}$$

In the two-phase region, we calculate the saturation of steam and water as follows:

$$S_s = \frac{\rho_w(h_w - h)}{h(\rho_s - \rho_w) - (h_s\rho_s - h_w\rho_w)}, S_w = 1 - S_s \tag{4.5}$$

4.2.2 Negative Compressibility

For the simultaneous solution, or fully coupled, fully implicit method, we solve for the pressure and enthalpy simultaneously using the mass (Equation 4.1) and the energy (Equation 4.2) conservation equations. Applying the chain rule to the accumulation terms in the mass (Equation 4.1) and energy (Equation 4.1) equations, we can reformulate these equations in terms of p and h to obtain:

$$V \begin{bmatrix} \frac{\partial \rho_t}{\partial p} & \frac{\partial \rho_t}{\partial h} \\ \frac{\partial \rho_t}{\partial p}h & \frac{\partial \rho_t}{\partial h}h + \rho_t \end{bmatrix} \begin{bmatrix} \frac{\partial p}{\partial t} \\ \frac{\partial h}{\partial t} \end{bmatrix} = \begin{bmatrix} 1 \\ H_{inj} \end{bmatrix} \Upsilon(p_{in} - p) \tag{4.6}$$

Eliminating $\frac{\partial h}{\partial t}$ from 4.6:

$$\gamma \frac{\partial p}{\partial t} = (1 - \beta H_{inj})\tau(p_{in} - p) \tag{4.7}$$

where:

$$\beta = \frac{\frac{\partial \rho_t}{\partial h}}{\frac{\partial \rho_t}{\partial h}h + \rho_t} \text{ and } \tau = \frac{\Upsilon}{V} \tag{4.8}$$

For two-phase cells we know from thermodynamic relationships (Figure 4.1):

$$\gamma = \frac{\partial \rho_t}{\partial p} - \beta \frac{\partial \rho_t}{\partial p}h < 0 \text{ (two-phase cells)} \tag{4.9}$$

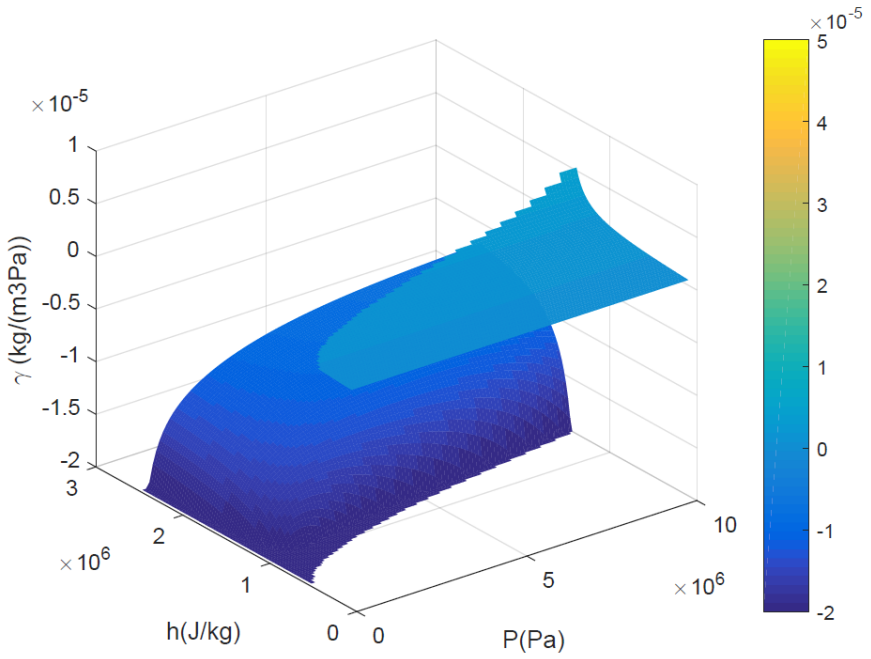
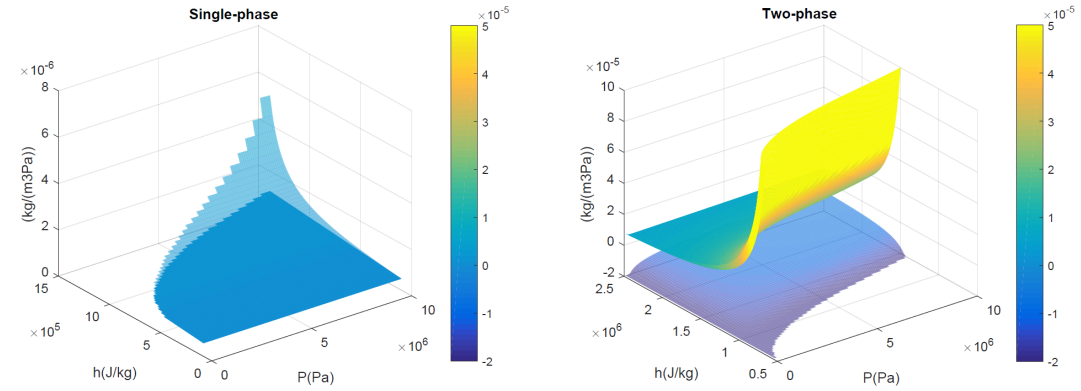


Figure 4.1: Plot of γ. The two-phase region (negative) is in blue; the single-phase liquid water region (positive) is in orange.

We find that the apparent compressibility, γ, is negative in the two-phase region (Figure 4.1). That is, when injecting cold water into the cell, the cell pressure decreases. Having an apparent negative compressibility makes it difficult for the nonlinear solver to converge. This is because we normally expect the container pressure to increase as the mass increases. Thus, depending on the size of the Newton update, we may obtain a negative pressure update. We will now go into detail of how a negative γ value can be problematic for a fully coupled, fully implicit method.

In Figure 4.2, we compare the apparent compressibility γ with $\left(\frac{\partial \rho}{\partial p}\Big|_{\partial h=0}\right)$. $\left(\frac{\partial \rho}{\partial p}\Big|_{\partial h=0}\right)$ is how we expect the density to change with a change in pressure at a fixed enthalpy. We notice that for single-phase cells, the apparent compressibility and $\frac{\partial \rho}{\partial p}\Big|_{\partial h=0}$ are both positive. This is as expected, because this means with an increase in density (mass increase), the pressure would also increase. However, for two-phase cells, we

see that although we still get the same increase in density at a fixed enthalpy, the apparent compressibility is negative. This difference in sign demonstrates how the inclusion of the energy equation results in this negative compressibility.



(a) Single-phase compressibility (b) Two-phase compressibility

Figure 4.2: Comparison of the apparent compressibility and actual compressibility (semitransparent surface is the apparent compressibility)

4.2.3 Fully Coupled, Fully Implicit Analysis

For the fully coupled and fully implicit backward Euler method, we solve for pressure and enthalpy for the current timestep $n+1$ simultaneously using the mass and energy conservation equations.

Accordingly, we discretize the problem in time:

$$\frac{\partial \rho_t}{\partial t} = \frac{\rho_t^{n+1} - \rho^n}{\Delta t}, \frac{\partial \rho_t h}{\partial t} = \frac{\rho_t^{n+1} h^{n+1} - \rho_t^n h^n}{\Delta t}, \tag{4.10}$$

and set all the other terms for timestep $n+1$. From Equation 4.1 and 4.2, the discrete form of the residual equation for this problem is:

$$R = \begin{bmatrix} R_M \\ R_E \end{bmatrix} = \begin{bmatrix} \frac{\rho^{n+1} - \rho^n}{\Delta t} - \tau(p_{in} - p^{n+1}) \\ \frac{\rho^{n+1} h^{n+1} - \rho^n h^n}{\Delta t} - H_{inj}\tau(p_{in} - p^{n+1}) \end{bmatrix} = 0 \tag{4.11}$$

To solve Equation 4.11, Newton's method ($J\delta x = -\Delta\omega R$, where $\Delta\omega$ is the size of the Newton step, $\Delta\omega = 1$ is the result of the full Newton step) is applied to linearize the system of nonlinear equations.

$$\begin{bmatrix} \frac{\partial R_M}{\partial p^{n+1}} & \frac{\partial R_M}{\partial h^{n+1}} \\ \frac{\partial R_E}{\partial p^{n+1}} & \frac{\partial R_E}{\partial h^{n+1}} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta h \end{bmatrix} = -\Delta\omega \begin{bmatrix} R_M \\ R_E \end{bmatrix} \tag{4.12}$$

$$\begin{bmatrix} \frac{\partial \rho_t}{\partial p} + \Delta t\tau & \frac{\partial \rho_t}{\partial h} \\ \frac{\partial \rho_t}{\partial p}h + \Delta t\tau H_{inj} & \frac{\partial \rho_t}{\partial h}h + \rho_t \end{bmatrix}^{n+1,k+1} \begin{bmatrix} p^{n+1,k+1} - p^{n+1,k} \\ h^{n+1,k+1} - h^{n+1,k} \end{bmatrix} = -\Delta\omega \begin{bmatrix} R_M \\ R_E \end{bmatrix}^{n+1,k} \tag{4.13}$$

The superscript k is the Newton iteration number. Applying the same procedure to obtain Equation 4.7 from Equation 4.6, we can eliminate $h^{n+1,k+1} - h^{n+1,k}$ to obtain the pressure update:

$$\left(\gamma^{n+1,k} + \Delta t(1 - \beta^{n+1,k}H_{inj})\tau\right)(p^{n+1,k+1} - p^{n+1,k}) = \Delta\omega(R_M^{n+1,k} - \beta R_E^{n+1,k})) \tag{4.14}$$

For the injection of cold water into saturated steam, we have:

$$\epsilon := (\beta H_{inj} - 1)\tau < 0 \tag{4.15}$$

Following these steps, we get the following pressure update:

$$p^{n+1,k+1} = p^{n+1,k} - \Delta\omega \frac{R_M^{n+1,k} - \beta R_E^{n+1,k}}{(\gamma^{n+1,k} - \Delta t\epsilon^{n+1,k})} \tag{4.16}$$

For the first Newton step, the standard practice is to use the solution of the previous timestep as the initial guess. Thus, we can write our initial Newton pressure update

as:

$$p^{n+1,k=1} = p^n - \Delta\omega \frac{R_M(p^n, h^n) - \beta R_E(p^n, h^n)}{(\gamma^n - \Delta t \epsilon^n)} \tag{4.17}$$

From Equation 4.17, we derive two critical timesteps for the fully coupled method [41]:

1. Δt_{neg}, the timestep where the pressure update in the full Newton step is negative. A timestep that is larger than t_{neg} would produce an update in the unphysical region, leading to a reduction in the timestep and repeating the Newton process.

2. Δt_{comp}, the timestep where the pressure update for the first Newton step diverges from the solution. A choice of timestep that is larger than t_{comp} would result in a nonlinear solution that diverges away from the final solution.

For full Newton updates, ($\Delta\omega = 1$) we obtain:

$$p^{n+1,k=1} = p^n - \frac{R_M(p^n, h^n) - \beta R_E(p^n, h^n)}{(\gamma^n - \Delta t \epsilon^n)} \tag{4.18}$$

and assuming $\Delta t < \frac{\gamma}{\epsilon^n}$:

$$R_M(p^n, h^n) - \beta(p^n, h^n) = \Delta t \epsilon^n (p_{in} - p^n) < 0 \tag{4.19}$$

because we are injecting ($p_{in} > p^n$ and $\epsilon > 0$), the updated pressure (Equation 4.18):

$$p^{n+1,k=1} < 0 \text{ if } p^n - \frac{\Delta t \epsilon^n (p_{in} - p^n)}{\gamma - \Delta t \epsilon^n} < 0 \tag{4.20}$$

Rearranging we obtain the timestep where the full Newton update pressure will be

negative:

$$\begin{aligned} p^n(\gamma - \Delta t\epsilon^n) &> \Delta t\epsilon^n(p_{in} - p^n) \\ \Delta t(-\epsilon^n p + \epsilon^n p - \epsilon^n p_{in}) &> -p\gamma \\ \Delta t_{neg} &= \frac{p^n\gamma}{\epsilon^n p_{in}} \end{aligned} \tag{4.21}$$

The first inequality is because γ is negative. Assuming that $\Delta t < \frac{\gamma}{\epsilon^n}$, the second inequality is from rearranging the terms. The final equality is the Δt such that the updated Newton pressure remains positive. We can see that if we take full Newton steps, the updated pressure will be negative, unless $\Delta t < \Delta t_{neg} < \frac{\gamma}{\epsilon^n}$. One can argue that one way to overcome this problem is to take damped Newton steps, $\Delta\omega < 1$; however, we can see that if $\Delta t > \frac{\gamma}{\epsilon^n}$ independent of $\Delta\omega$:

$$p^{n+1,k+1} > p^{n+1,k}, \text{ since } -\frac{\Delta t\epsilon^n(p^n - p_{in}}{\gamma - \Delta t\epsilon^n} > 0$$

Thus, the computed answer will diverge from the true solution. It is then clear that even if extremely small Newton steps are taken, the Newton path will diverge away from the true solution. We will define this diverging timestep as:

$$\Delta t_{comp} := \frac{\gamma}{\epsilon}$$

4.2.4 Newton Path

We next demonstrate the behavior of the Newton path as a function of the timestep size. Each Newton path shows how the nonlinear solution evolves based on the choice for the initial guess [79]. In Figure 4.3, we see that for a timestep $t < t_{neg}$ and using the previous timestep solution as the initial guess produces a Newton path that is close to linear.

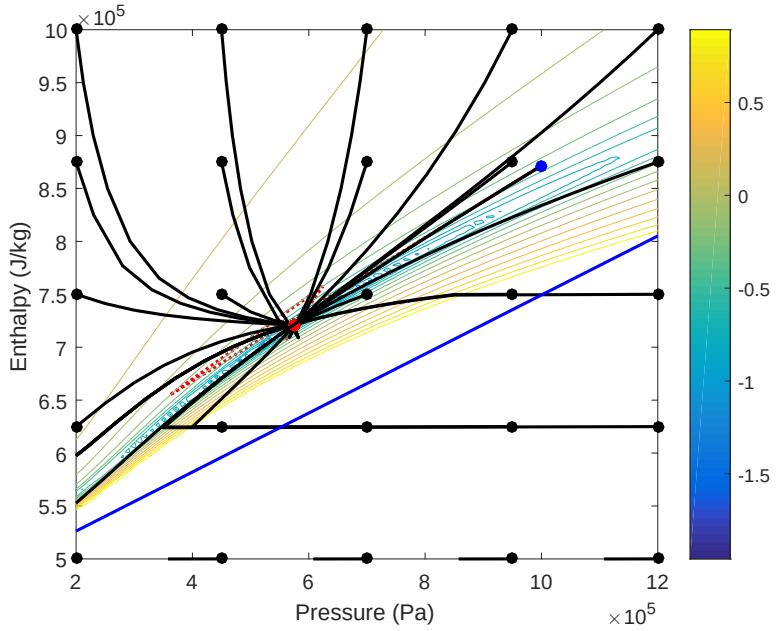
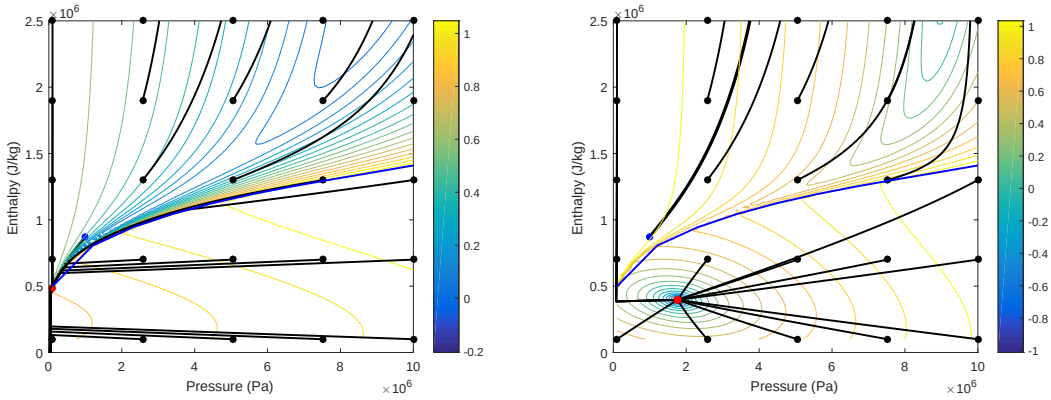


Figure 4.3: Path of Full Newton for converged case ($t < t_{neg}$). Residual contours ($\log_{10} L_2$-norm), the solid blue line corresponds to the phase boundary. The red dot shows the solution at that timestep, the black dots show different initial guess, the solid black lines are the Newton paths for the respective initial guess. The solid blue line is the phase boundary. The blue dot is using the solution at the previous timestep

However, if we choose a timestep $t > t_{neg}$, we have two different cases. The case where $t_{neg} < t < t_{comp}$ is shown in Figure 4.4a, and the case where $t > t_{comp}$ is shown in Figure 4.4b. For both cases, there is part of the solution space, where the Newton paths diverge completely from the solution. However, in Figure 4.4a the previous solution (blue dot) does not include this diverging region.

For both timesteps, it can be seen that if the initial guess is in the single-phase region, all the Newton paths converge to the solution (Figure 4.4a and 4.4b). For this single-cell problem, if the initial guess is in the single-phase region, the Newton method will converge.



(a) Newton Paths for $t_{neg} < t < t_{comp}$ (b) Newton Paths for $t > t_{comp}$

Figure 4.4: Residual contours ($\log_{10} L_2$-norm) and Newton paths for $t_{neg} < t < t_{comp}$ and $t > t_{comp}$. The red dot shows the solution for our timestep; the black dots show different initial guess; the solid black lines are the Newton paths for the respective initial guess. The solid blue line is the phase boundary. The blue dot indicates using the previous timestep solution as the initial guess.

4.2.5 Sequential Fully Implicit

For the sequential fully implicit (SFI) formulation, the mass and energy conservation equations are solved sequentially for their respective primary variables. When each equation is solved, a coupling variable is defined. The coupling variable is the property that is fixed while the subproblem (mass or energy equation) is solved. The pressure-enthalpy formulation does not require any variable switching; thus, it allows for a simpler implementation and analysis of this SFI formulation. It is important to note that if the sequential formulation is stable and convergent, the resulting numerical solution will be identical to the fully coupled solution up to the desired tolerance. Algorithm 1 describes the SFI scheme.

Step 1: Solve:

$$R_M^{n+1}(p^{n+1}, \partial h = 0)) = 0 \tag{4.22}$$

The coupling variable for the flow step is enthalpy, which is fixed while solving the

flow equations.

Step 2: Solve:

$$R_E^{n+1}(p(\partial\rho = 0), h^{n+1}) = 0 \tag{4.23}$$

This involves solving the energy balance with a fixed density. Steps 1 and 2 are repeated until convergence. The use of a fixed density while solving the energy equation is convergent for two-phase cells. Note that for two-phase cells, keeping the pressure constant while solving the energy equation is unconditionally divergent [78].

Algorithm 4.1 Sequential Fully Implicit

$p^\nu = p^n, h^{\eta^*} = h^n$

while $|||[R_M; R_E]||_2 > \epsilon_R$ **do**

  while $||R_M||_2 > \epsilon_M$ **do** (Step 1)

    $dp = -J_M(p^\nu, h^{\eta^*})^{-1} R_M(p^\nu, h^{\eta^*})$

    $p^{\nu+1} = p^\nu + dp$

  end while

  $p^{\nu^*} = p^{\nu+1}$

  $\rho^{\nu^*} = \rho(p^{\nu^*}, h^{\eta^*})$

  while $||R_E||_2 > \epsilon_E$ **do** (Step 2)

    $dh = -J_E^{-1}(c_T^{\nu^*}, h^\eta) R_E(p^{\nu^*}, h^\eta)$

    $h^{\eta+1} = h^\eta + dh$

  end while

  $h^{\eta^*} = h^\eta$

end while

Here:

- ν is the index for the Newton iteration in the mass conservation equation, and μ^* is the index for the converged solution for the mass equation

- η is the index for the Newton iteration in the energy conservation equation, and η^* is the index for the converged solution for the energy equation
- J_M is the Jacobian for the mass residual equation
- J_E is the Jacobian for the energy residual equation
- ϵ_R is the convergence criterion for both the mass and energy residuals
- ϵ_M is the convergence criterion for the mass residual
- ϵ_E is the convergence criterion for the energy residual

The nonlinear sequential solution for the single-cell case is as follows:

$$R_M = \frac{\rho_t^{n+1} - \rho_t^n}{\Delta t} - \tau(p_{in} - p^{n+1}) = 0 \tag{4.24}$$

$$R_E = \frac{(\rho_t h)^{n+1} - (\rho_t h)^n}{\Delta t} - H_{inj}\tau(p_{in} - p^{n+1}) = 0 \tag{4.25}$$

In the SFI formulation, we first solve for pressure using the mass residual equation assuming a fixed enthalpy. Linearizing R_M using Newton's method we obtain:

$$p^{n+1,k+1} = p^{n+1,k} - \frac{R_M(p^{n+1,k}, h^{n+1,k})}{\frac{\partial R_M}{\partial p}\Big|_{\partial h=0}(p^{n+1,k}, h^{n+1,k})} \tag{4.26}$$

Suppose we use the solution from the previous timestep as the initial guess for the first sequential iteration, that is, $h^{n+1,k=0} = h^n$ and $p^{n+1,k=0} = p^n$, then the residual for the mass equation of the first Newton step can be written as:

$$R_M(p^n, h^n) = -\tau(p_{in} - p^n)\Delta t < 0 \tag{4.27}$$

We also know from thermodynamic relationships that the gradient for this residual is:

$$\left.\frac{\partial R_M}{\partial p}\right|_{\partial h=0} = \left.\frac{\partial \rho_t^{n+1}}{\partial p}\right|_{\partial h=0} + \tau > 0 \tag{4.28}$$

Thus, we have for the first Newton iteration in the flow sequential subloop and $p^{n+1,k=0} = p^n$ that:

$$p^{n+1,k=1} = p^n + \frac{\tau(p_{in} - p^n)}{\left.\frac{\partial \rho_t^{n+1}}{\partial p}\right|_{\partial h=0} + \tau} > p^{n+1,k} \tag{4.29}$$

and the increase in the pressure is independent of the timestep size. Because we fix the enthalpy in the mass sequential sub-loop, a phase change will occur if $p^{n+1,k} > p_{sat}(h^n)$; that is, the phase change takes place if the updated pressure is greater than the saturation pressure for a (saturated) liquid water enthalpy of h^n. Thus, the fluid in the cell condenses for the case where:

$$p^n + \frac{\tau(p_{in} - p^n)}{\left.\frac{\partial \rho_t^{n+1}}{\partial p}\right|_{\partial h=0} + \tau} > p_{sat}(h^n) \tag{4.30}$$

Rearranging these equations, we have the following inequality:

$$\frac{\tau(p_{in} - p^n)}{\left(\left.\frac{\partial \rho_t^{n+1}}{\partial p}\right|_{\partial h=0} + \tau\right)(p_{sat}(h^n) - p^n)} > 1 \tag{4.31}$$

For the single-cell problem, we are able to obtain a strict bound on when the sequential solution will condense. We can also rewrite this inequality in terms of the residual and Jacobian (Equation 4.26) to give:

$$C_{neg} = \frac{R_M(p^n, h^n)}{\frac{\partial R_M}{\partial p}|_{\partial h=0}(p^n, h^n)\,(p_{sat}(h^n) - p^n)} > 1 \tag{4.32}$$

Step 2: Using the pressure update, we solve the energy conservation equation for

enthalpy using R_E, such that $\partial\rho_T = 0$. That is:

$$h^{k+1} = \frac{\rho_t^n h^n + \tau H_{inj}(p_{in} - p^{n+1,k+1})}{\rho_t(p^{k+1}, h^k)} \tag{4.33}$$

We can see that for the sequential single-cell model, the updates for enthalpy are linear. Although $\rho(p, dh = 0)$ is a nonlinear function of pressure, this nonlinearity is quite mild.

4.2.6 Comparison between Fully Coupled and Sequential Fully Implicit solutions

In this section, we compare the fully coupled and sequentially coupled solutions for the single-cell problem for a range of timestep sizes. We see from Figure 4.5 that for $\Delta t = 0.15$ days, the water condenses fully. As expected, when the fluid in the cell has condensed completely into liquid water, the pressure begins to increase. This is because the cell is no longer in the two-phase region, and liquid water has a small positive compressibility. This results in an inflection in the pressure solution after the phase change. We see that for the fully coupled problem, the full Newton pressure update becomes negative at a timestep of 0.15 days. However, for the sequential formulation, the first update of the pressure is always positive after a timestep of 0.15 days and is close to the final solution. However, prior to this time, the fully coupled solution provides a better initial guess to both the saturation and pressure compared with the sequential method.

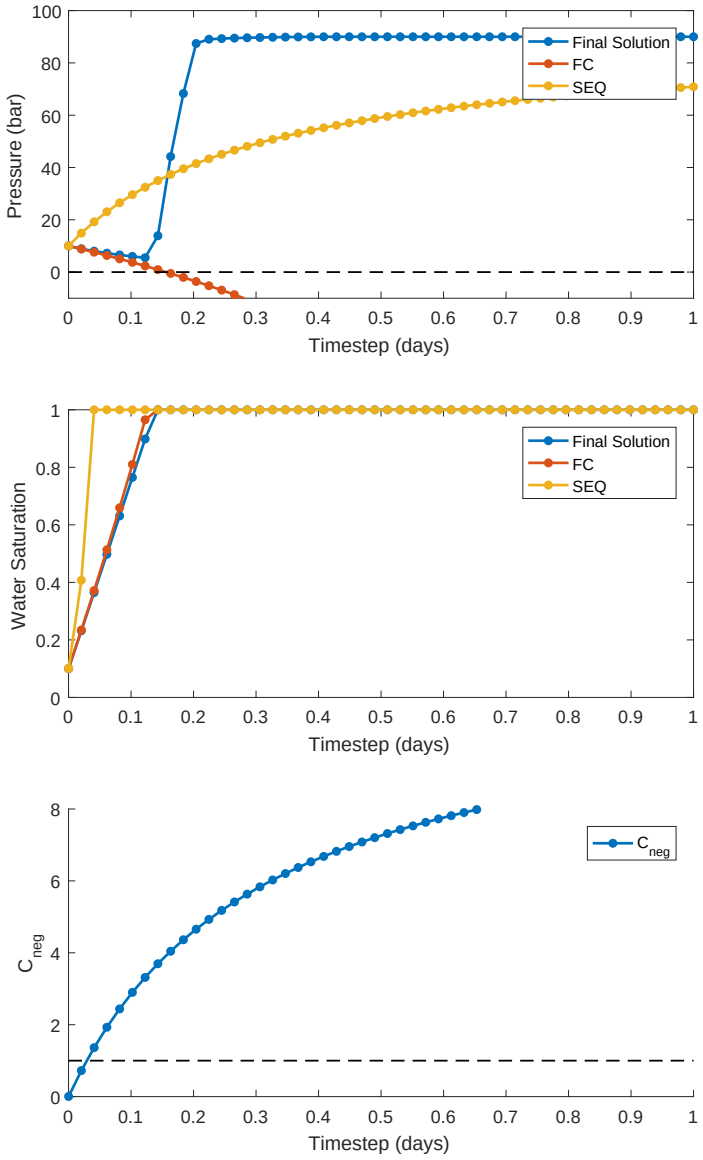


Figure 4.5: Initial Newton update for different timesteps

From Figure 4.5, two main conclusions can be drawn. First, we see that for timesteps smaller than 0.15 days, the fully coupled problem converges. However, the sequential scheme is convergent for timesteps greater than 0.15 days, but it is not convergent for timesteps smaller than 0.15 days. Here, we find that the full

coupled formulation converges for the cases where it does not condense fully, and the sequentially coupled formulation converges for the the cases where condensation takes place.

One way of utilizing both formulations for the appropriate timesteps would be to first run the fully coupled problem, and if the first Newton update is negative and does not converge, we can switch to the sequential scheme. From the results, we see that this guarantees convergence for all timestep sizes. This is the underlying idea for the 'nonlinear' preconditioner for multicell simulations, which is described in detail in Section 4.3

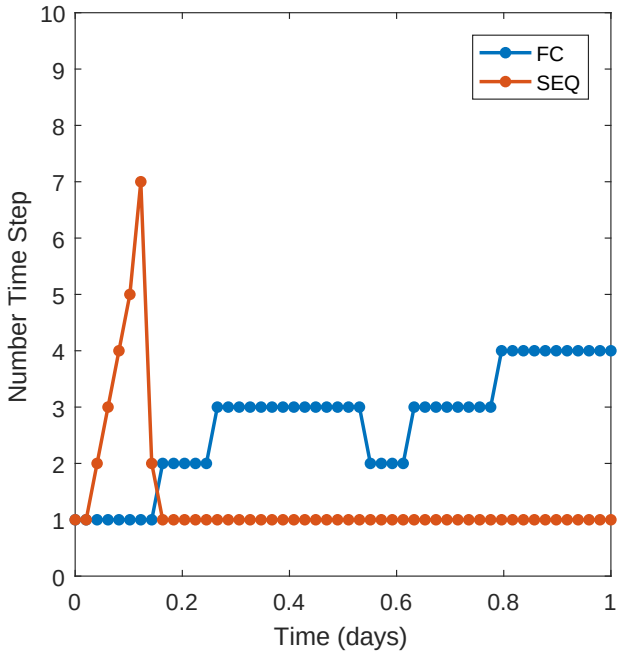


Figure 4.6: Component CFL of converged solution for fully coupled and sequential fully implicit. Black dashed line indicates the timestep where the solution fully condenses to single-phase liquid

Newton Paths

Here, we compare the Newton paths (Figure 4.7) for the same timestep as shown in Figure 4.4b for the sequential formulation. As previously shown, for this timestep, the fully coupled solution diverges away from the solution (thick solid black line originating from blue dot). We see that using the previous time step solution (blue dot) as the initial guess for the sequential approach (red dotted line), we converged to the final solution (red dot). We also include the case where we use the solution of the first mass solution (blue diamond) in the sequential formulation as an initial guess for the fully coupled problem. We see that for this case because the initial guess is in single-phase, it converges to the final solution (black dotted line). Using the first solution of the mass conservation equation as an initial guess is the foundation of the preconditioner that we describe in the Section 4.3.

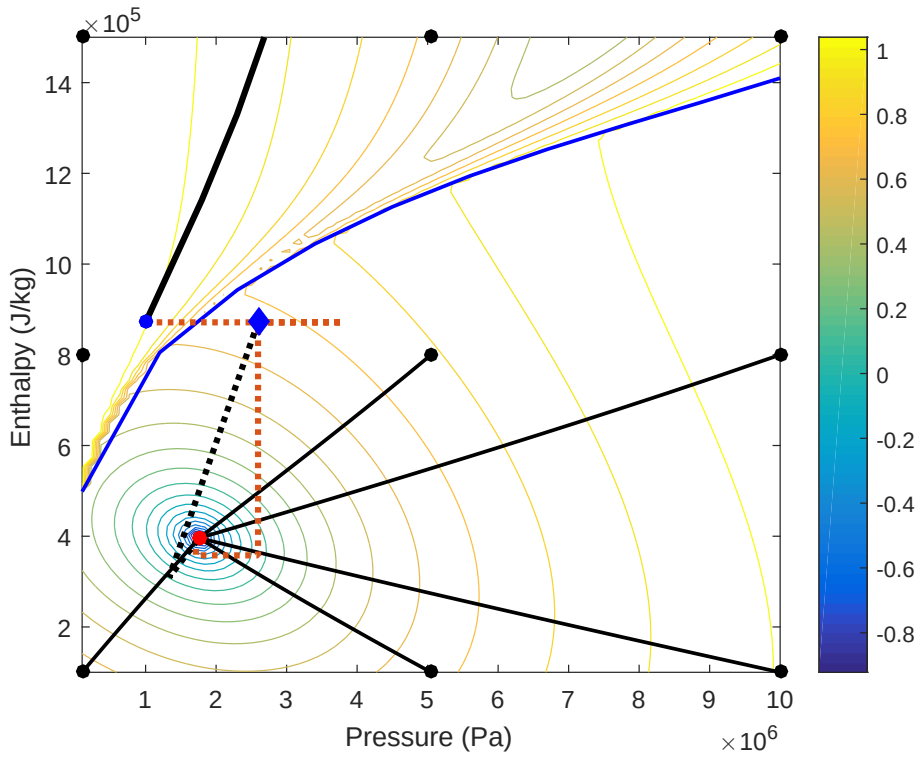


Figure 4.7: Path of the Newton path for a sequential formulation (same timestep as Figure 4.4b) (Red dotted line). The blue diamond is the initial guess resulting from the sequential preconditioning step. The black dotted line is the fully coupled Newton path using the blue diamond as the initial guess. Residual contours ($\log_{10} L_2$-norm), the solid blue line corresponds to the phase boundary. The red dot shows the solution at that timestep, the black dots show different initial guess, the solid black lines are the Newton paths for the fully coupled problem for the respective initial guess. The solid blue line is the phase boundary. The blue dot is using the solution at the previous timestep.

4.3 Modified Sequential Fully Implicit Preconditioner (m-SFI-P)

In the previous section, we showed that the fully coupled method was converging for the case where the cell does not fully condense, and the sequential fully implicit

formulation converges for the solution where it does condense. In order to take advantage of these two opposite effects, we developed the m-SFI-P where the mass equation is first solved for all the two-phase cells. Using this solution as an initial guess, we can use the fully coupled method to iterate to convergence. As shown in Figures 4.4a and 4.4b, we found that for the single-cell case, if an initial guess in the single-phase region is used, the fully coupled problem converges. In this section, we present the algorithm and the results of how this preconditioner generalizes to multicell problems in one and two dimensions.

4.3.1 Algorithm

The m-SFI-P step is a nonlinear preconditioner that computes a good initial guess for the fully coupled, fully implicit, Newton method. From the single-cell model, we see that the fully coupled, fully implicit method struggles to converge for large timesteps when there is steam condensing due to the injection of cold water. This preconditioning step aims to provide an initial condition where we have a good estimate of where the condensation front is (Figure 4.7).

The preconditioning step entails solving the discrete nonlinear residual mass conservation equations for all two-phase cells. The classification of the two-phase cells is updated after each Newton iteration. Specifically, if a cell has condensed during the previous Newton iteration, it is no longer in the domain to be solved during this step. The algorithm is described in Algorithm 4.2.

One key step in the m-SFI-P scheme is that the enthalpy remains fixed for all the cells. We determine whether there is a phase change, if the saturation of water is greater than unity. If this is the case, the single-phase water temperature, T, is calculated based on the enthalpy of the previous timestep. That is:

$$\text{Phase update} = \begin{cases} \text{Single-phase} & T^{n+1,k+1} = T(p^{n+1,k+1}, h^n) \\ \text{Two-phase} & S_w^{n+1,k+1} = S_w(p^{n+1,k+1}, h^n) \end{cases}$$

Algorithm 4.2 m-SFI-P

if $C_{neg} > 1$ **then**

$p^\nu = p^n$

while $||R^{\tilde{M}}||_2 > \epsilon_M$ **do** (Step 1: Solve the mass conservation equation for two-phase cells)

$d\tilde{p} = -J_{\tilde{M}}(p^\nu, h^n)^{-1} R_M$

$p^{\nu+1} = p^\nu + dp$

Update $\tilde{M}(p^{\nu+1})$ (Update phase state of each cell)

end while

$p^{k=0} = p^{\nu*}$ (Use m-SFI-P solution as initial guess for fully coupled)

$h^{k=0} = h^n$

while $||R^M, R^E||_2 > \epsilon_E$ **do** (Step 2 Fully Coupled Loop)

$dx = -J^{-1} R$

$x^{k+1} = x^k + dx$

end while

end if

- $J_{\tilde{F}} = \frac{\partial R_m}{\partial p}|_h$ is the Jacobian from the mass residual for all the two-phase cells
- $\tilde{F}_i$ is the flow residual for two-phase cells
- p^ν is the pressure solution for the ν iteration in the preconditioning step and $p^{\nu*}$ is the final preconditioned solution
- J, R is the Jacobian and residual from the fully coupled method

- x^k is the vector of the primary variables of the solution for the fully coupled method for iteration k

4.3.2 Criterion for m-SFI-P

Here, we show the derivation of a bound that indicates when this preconditioner will be effective. The purpose of the preconditioner is to provide a good initial guess of the condensation front by solving the mass equation. This prevents the unnecessary calculation of this preconditioning step. We know that condensation will occur if the pressure update is larger than the difference between the current pressure and the saturation pressure. Unlike the single-cell problem, we are unable to obtain this pressure update exactly without solving the full system of equations. However, we can exploit certain properties of the residual and Jacobian to obtain a bound of when the preconditioner will not be effective.

The Jacobian for a single parabolic diffusivity-type equation for any rectangular mesh has been shown to be diagonally dominant [52]. Here, the Jacobian is from the mass residual equation of two-phase cells.

$$J\delta p = -R_M, J = \frac{\partial R_M}{\partial p} \tag{4.34}$$

A known result of diagonally dominant matrices from [67] is that:

$$||J^{-1}||_\infty \leq \max_{1\leq i\leq n} \frac{1}{\Delta_i(J)} \tag{4.35}$$

where:

$$\Delta_i(J) = |J_{ii}| - \sum_{j\neq i} |J_{ij}| \geq 0 \tag{4.36}$$

and $||\cdot||_\infty$ is the ℓ_∞ matrix norm

For this case, we know that the diagonals of the Jacobian are the sum of the

derivatives of the accumulation and flux terms with respect to pressure for the given cell. The off-diagonal terms are the derivatives of the flux terms. Thus, for $\Delta_i(J)$ we only have the derivative of the accumulation term with respect to pressure.

$$\Delta_i(J) = |J_{ii}| - \sum_{j \neq i} |J_{ij}| = \left| \frac{\partial \rho_i}{\partial p_i} \right|_{\delta h = 0} \tag{4.37}$$

From Equations 4.34, 4.35, 4.36 and 4.37 we can obtain the bound:

$$\frac{||\delta p||_\infty}{||R_M||_\infty} = \frac{||J^{-1} R_M||_\infty}{||R_M||_\infty} \leq ||J||_\infty \leq \max_{1 \leq i \leq n} \frac{1}{\Delta_i(J)} = \max_{1 \leq i \leq n} \frac{1}{\left| \frac{\partial \rho_i}{\partial p_i} \right|_{\delta h = 0}} \tag{4.38}$$

Thus, we know that the largest pressure change in a cell $||\delta p||_\infty$:

$$||\delta p||_\infty \leq \frac{||R_M||_\infty}{\min_i \left| \frac{\partial \rho_i}{\partial p_i} \right|} \tag{4.39}$$

We can use this as an indicator of whether a cell will condense or not. We know that if the update in pressure is small enough then the cell will change phase specifically if:

$$\min_i \left[p_{sat}(h_i^n) - p_i^n \right] \leq ||\delta p||_\infty \leq \frac{||R_M|| \infty}{\min_i \left| \frac{\partial \rho_i}{\partial p_i} \right|} \tag{4.40}$$

where i is the cell number, and p_i^n and h_i^n is the pressure and enthalpy at the previous timestep. If the bound (Equation 4.40) holds then we know that for the m-SFI-P step it is possible for condensation to occur, since the pressure change is greater than the smallest pressure difference. Thus we can precompute this value C_{neg} value over the cells:

$$C_{neg} := \frac{||R_M||_\infty}{\min_i \left| \frac{\partial \rho_i}{\partial p_i} \right| \min_i \left[p_{sat}(h_i^n) - p_i^n \right]} \geq 1 \tag{4.41}$$

4.4 AD-GPRS

The general sequential framework in AD-GPRS is the foundation to the implementation of the m-SFI-P method. The m-SFI-P method can be formulated as a sequential method where the m-SFI-P step is solved before the fully coupled method. This allows the m-SFI-P solution to be the initial guess for the fully coupled method. To specify this nonlinear solver we utilize the mapper construct that is available in the GENIC. The mappers allow for a general specification of which cells to include at each solution step. Every subproblem has a default mapper of including all cells.

```
COUPLING
SEQ(FLOW<MAPPER_T3>,FIM(FLOW<MAP_FULL>,THERMAL<MAP_FULL>),2) /
/
```

Figure 4.8: AD-GPRS Input File Example for Sequential Preconditioner

We specify these mappers with the COUPLING keyword, this is shown in Figure 4.8. The key to using AD-GPRS to describe this preconditioner is that we specify a sequential-implicit scheme that iterates only for one sequential loop. The first step is the preconditioning step where we solve the flow equations for FLOW<MAPPER_T3>. Here, the <MAPPER_T3> prescribes that if Equation is satisfied, then all two-phase cells will be solved at that step. The determination of the two-phase cells is based on the current Newton iteration, this means that cells could be added or removed to the subdomain <MAPPER_T3> during the solution step of FLOW<MAPPER_T3>.

Once this step has converged, the converged solution of FLOW<MAPPER_T3> is fed as the initial guess to the fully coupled approach. If this step does not converge, then the entire algorithm does not converge and a time step cut occurs. But if the step does converge it loops back and does one final check of all the residuals. But because

we know that the solution to the fully coupled problem converged, the preconditioning step should converge, because it is a subset of the fully coupled solution.

4.5 Nonlinear Analysis for One-dimensional Simulation

4.5.1 Problem Description

Here, we look at how the m-SFI-P approach compares with the fully coupled formulation for a one-dimensional test problem. For all the cases, we selected the initial timestep to be the total time and a timestep multiplier of four. We used the same boundary and initial conditions as the single-cell problem. The AD-GPRS input file for the 10 cell model can be found in Figure A.4.

- One boundary cell at P: 90 bar, T: 355K
- Initial conditions 10 bar, 455K, $S_g = 0.9$

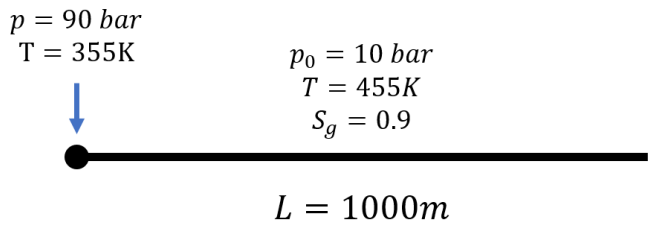


Figure 4.9: Schematic of one-dimensional negative compressibility problem

4.5.2 Nonlinear solution

To demonstrate the effectiveness of the m-SFI-P, we present the Newton solution for a single timestep for a Courant-Friedrich-Lewy (CFL) value of 2.96. Here, the CFL

number is defined as:

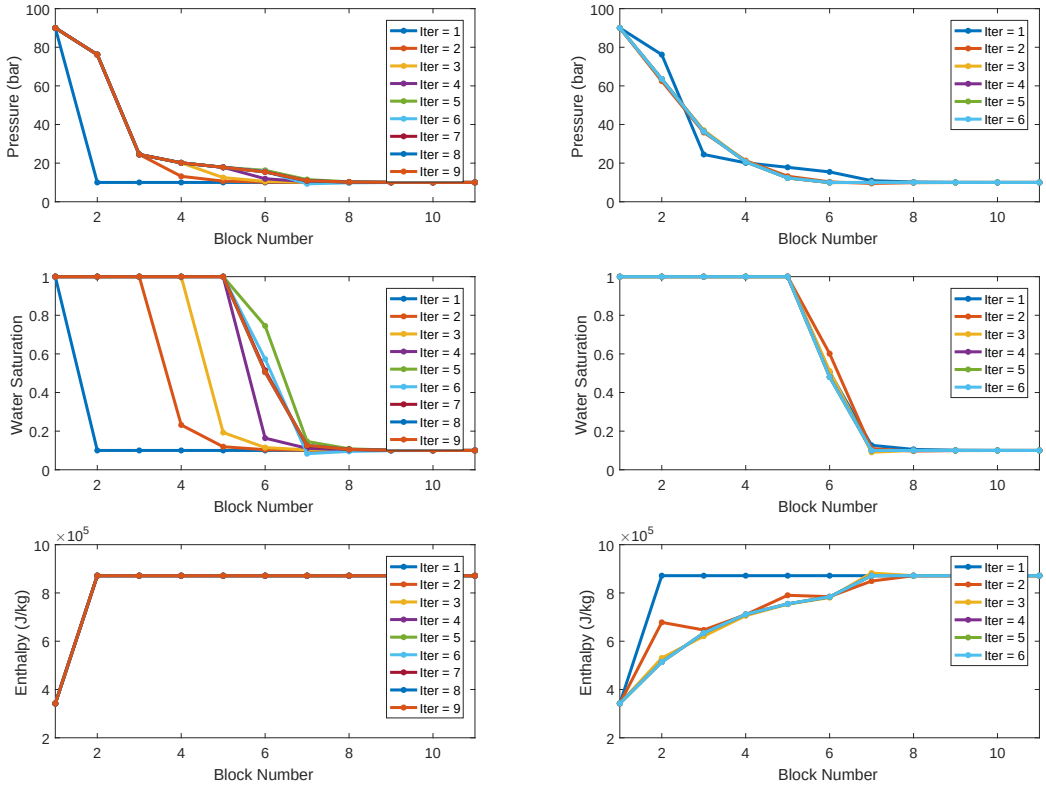
$$\text{CFL} = \frac{q_T \Delta t}{V \phi} \tag{4.42}$$

here:

- q_T is the total volumetric rate through the grid cell
- V is the volume of the grid cell
- Δt is the timestep
- ϕ is the porosity of the grid cell

For this timestep, the standard fully coupled fully implicit formulation does not converge for the first Newton step; as a result, m-SFI-P is used.

As expected, we see that in Figure 4.10a in the m-SFI-P step, the enthalpy is constant for all the Newton iterations, because this is fixed when solving the mass equation. We can see that with the m-SFI-P the fully coupled steps, the saturation front remains at approximately the same position and the pressure and enthalpy fronts ultimately smoothen out.



(a) Newton solutions for m-SFI-P step (b) Newton solutions after m-SFI-P step

Figure 4.10: Pressure (top), Water Saturation (middle) and Enthalpy (bottom) for different Newton iterations in the m-SFI-P step and in the fully coupled step after m-SFI-P

4.5.3 m-SFI-P Criterion for One-dimensional Simulation

In this section, we demonstrate the effectiveness of m-SFI-P for C_{neg} values that satisfy or violate the prescribed bound. We demonstrate this through two different models with two different initial conditions shown in Figures 4.11 and 4.12. The initial saturation is identical in both initial conditions (left and right), with different initial pressure conditions. For the case where the m-SFI-P works, there is a sharp change in the pressure at the front from 90 bar to 10 bar. But for the case where the

m-SFI-P does not work, the pressure solution does not change.

What we observe is that when there is an abrupt change in the pressure, the m-SFI-P step is able to obtain a good initial guess of where the saturation front is. However, when the pressure is relatively smooth, and the initial and final solutions do not vary much in the pressure solution, the m-SFI-P does not work well. What we observe that is common in both the fully coupled and m-SFI-P case is that the pressure wants to vary where the final solution does not have much of a change in the pressure.

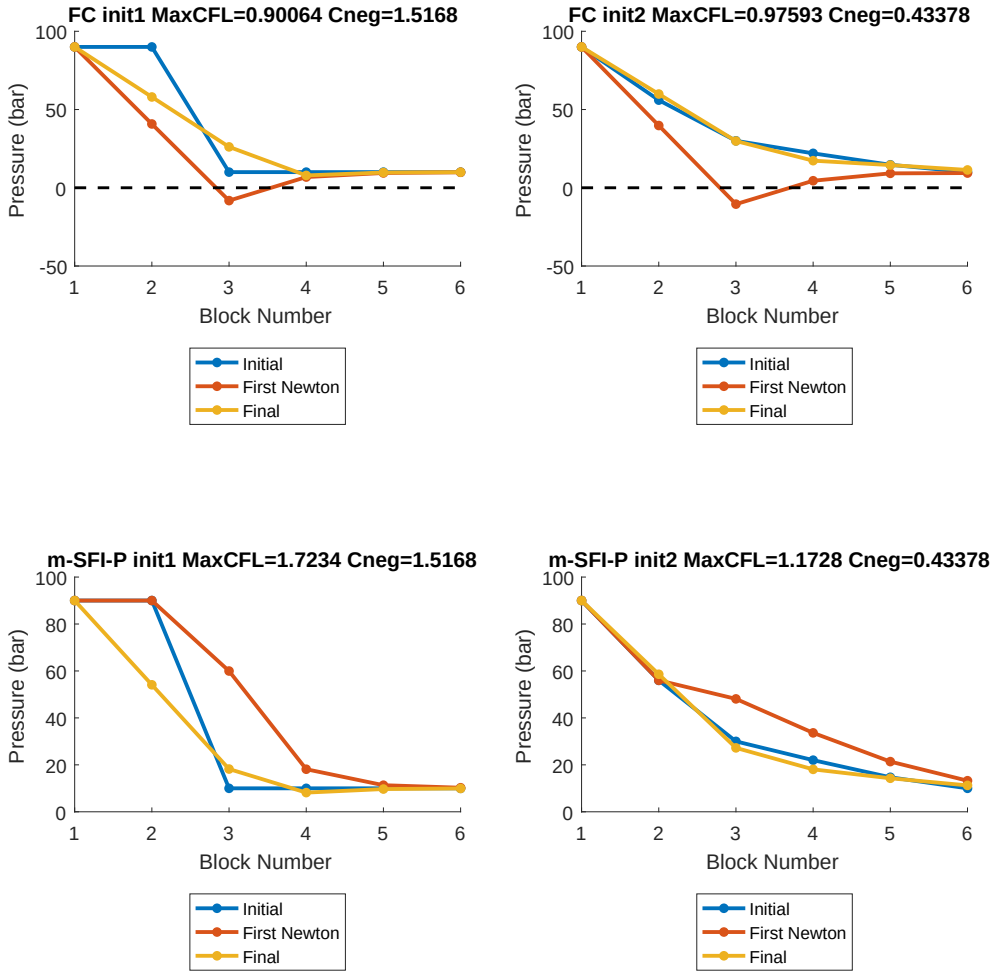


Figure 4.11: Comparison of the first Newton update for pressure for fully coupled and m-SFI-P (top and bottom) for two different initial conditions (left and right)

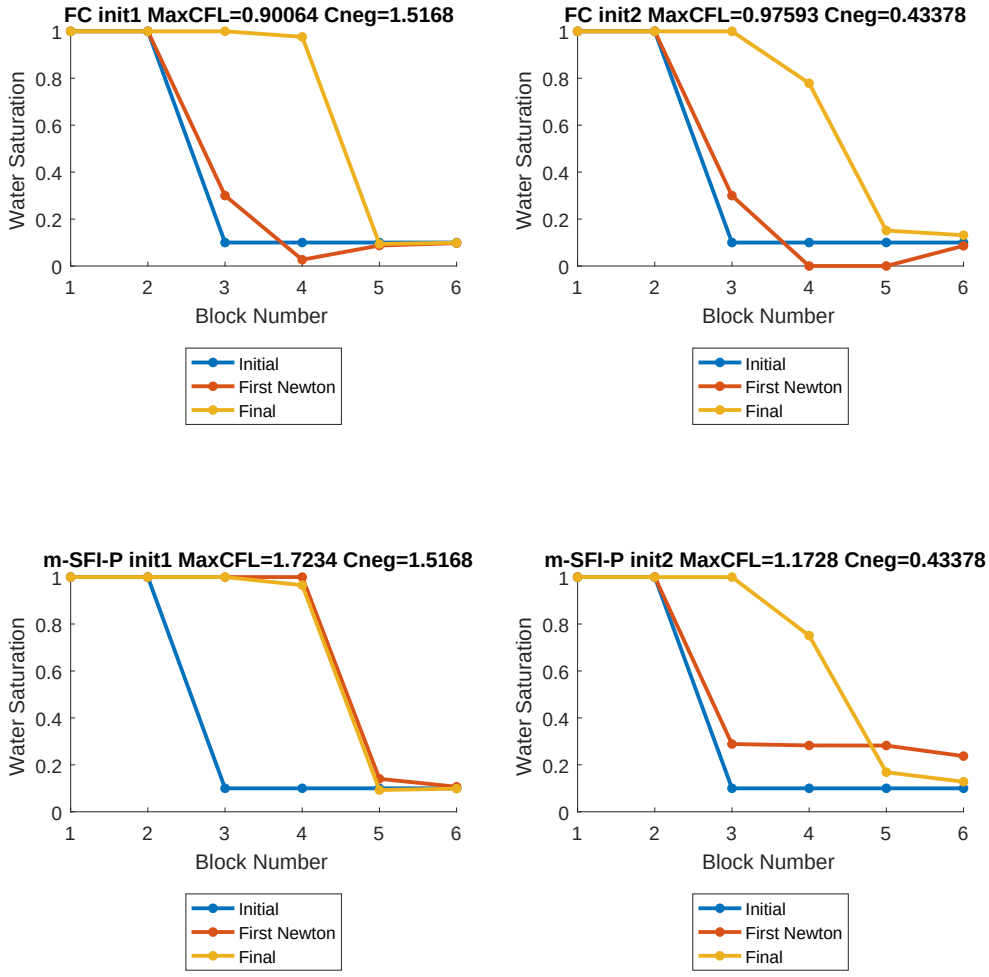


Figure 4.12: Comparison of first Newton update for saturation for fully coupled and m-SFI-P (top and bottom) for two different initial conditions (left and right)

4.5.4 Refined Solution

Before investigating the nonlinear solution of the m-SFI-P approach, we will first demonstrate how this negative compressibility issue is a numerical issue due to the

discretization of the problem. To demonstrate this, we ran two simulations, one with 5 cells and another with 50 cells. Using the results of the 50 cell model, we computed the cell-averaged pressure on the 5 cell grid. We see that the cell-averaged pressure does not decrease but actually increases. However, we see that for the refined solution there is still a slight decrease in pressure solution right at the two-phase front. This is because in the sharpness of the front is limited by the grid refinement. If we compare the m-SFI-P solution and the fully coupled first Newton step for this timestep, we see that the m-SFI-P imitates the refined solution by increasing the pressure.

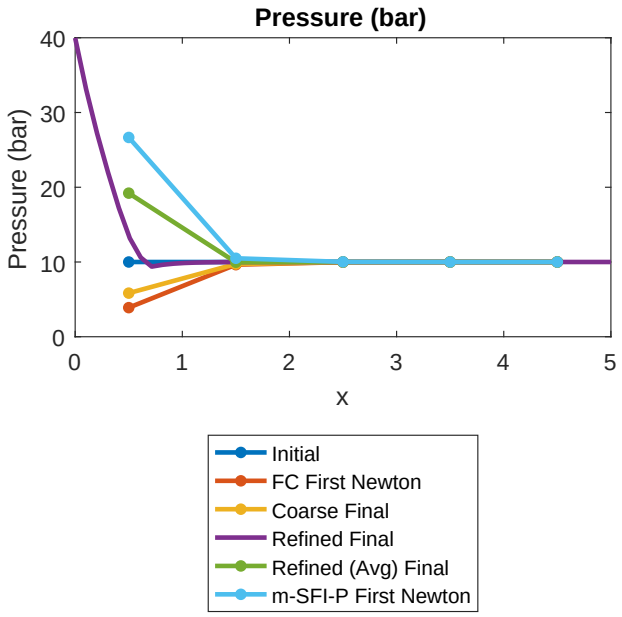


Figure 4.13: Comparison of pressure solution for the refined problem and fully coupled and m-SFI-P solutions for timestep of 0.05 days. (CFL 0.846 on the coarse grid)

4.5.5 Refinement in Space and Time

In order to examine the performance of the m-SFI-P with grid refinement, we looked at the maximum converged CFL for the fully coupled method and the m-SFI-P approach.

We found that as we refined the problem, the m-SFI-P method performed better than the fully coupled for all resolutions. For the fully coupled method, the maximum CFL for all grid sizes was limited to values ranging from 3 to 4. However, for the m-SFI-P, the maximum CFL scales with the number of cells in the model. This is because as we refine the problem, the cell size decreases, so the CFL to completely fill the entire domain with cold water increases.

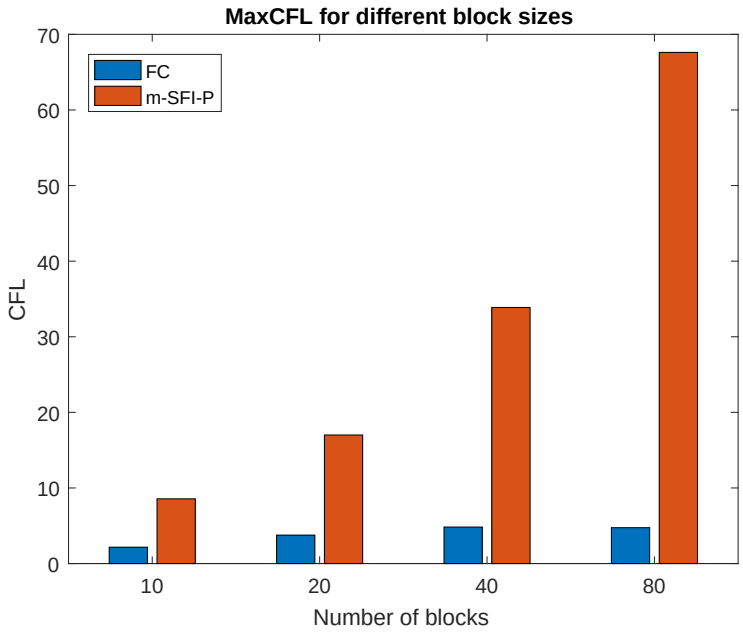


Figure 4.14: Maximum CFL for varying number of cells

4.6 Two-dimensional Results

The distribution of permeability and porosity were taken from the top layer of the SPE10 model [8]. This distribution can be seen in Figure 4.15. There are $220 \times 60 \times 1$ cells, each grid cell is $10 \times 20 \times 70$m. Uniform pressure (10 bar) and temperature (455.15K) and water saturation (0.2) was imposed as the initial conditions. There is a single constant pressure source in the model (T=$355K$, p = 90 bar)

From Figure 4.17, we see that at the initial timesteps, the m-SFI-P approach is able to succeed with CFL numbers that are one to two orders of magnitude larger than the fully coupled model. This numerical experiment shows how this preconditioner is able to generalize to heterogeneous multidimensional models.

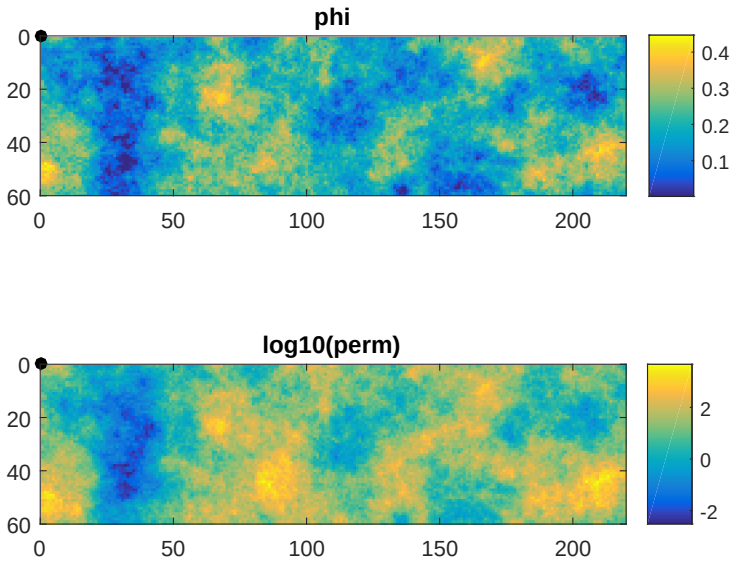


Figure 4.15: Porosity and permeability distribution

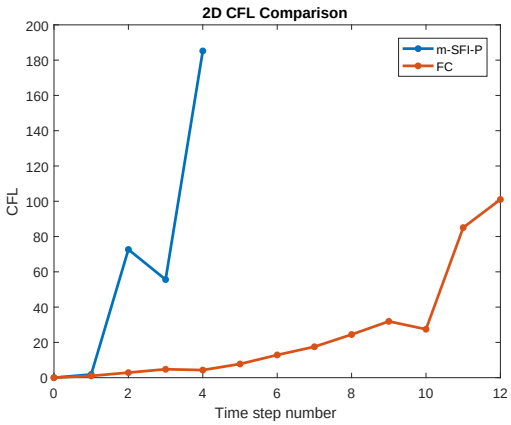


Figure 4.16: Component CFL of converged solution

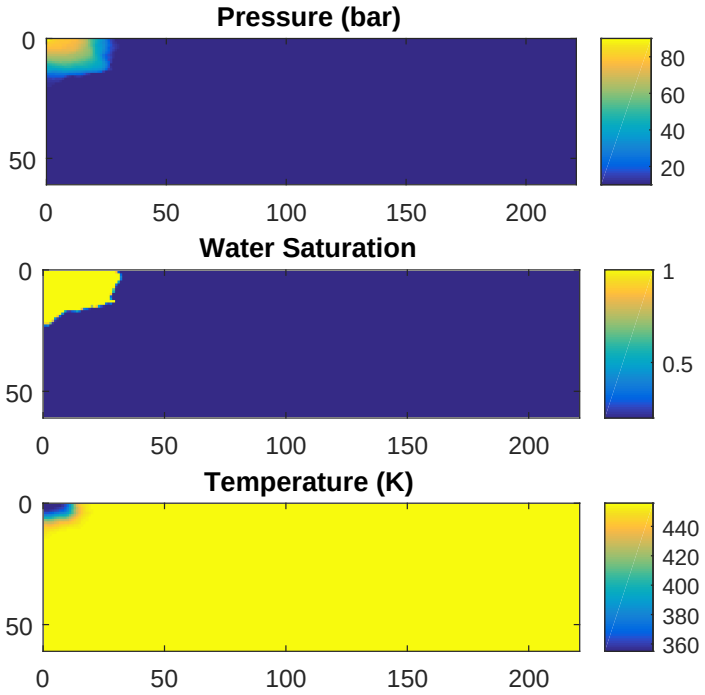


Figure 4.17: Final solution

4.7 Chapter Summary

One of the main sources of difficulty in geothermal reservoir simulation is the effect of the apparent "negative compressibility" associated with computational cells that have both liquid and vapor. This negative compressibility presents a significant difficulty for the nonlinear solver when a simultaneous solution strategy is used for solving the mass and energy equations. We analyzed this steam condensation problem using a single-cell problem. These issues include Newton iterations in nonphysical regions and divergence of the Newton paths away from the correct solution.

A nonlinear preconditioner referred to as a Modified Sequential Fully Implicit Preconditioner (m-SFI-P) was proposed. A criterion was also developed to indicate

when this preconditioner would provide a good initial guess to the fully coupled method. This preconditioner removed a limit on the CFL values for which a typical fully coupled method was convergent. The effectiveness of the m-SFI-P was sustained even with refinement for a one-dimensional problem.

In essence, this preconditioner allows the near-elliptic pressure solution from the mass conservation equation to first determine where the phase front should be. Using this as an initial guess, the fully coupled problem was able to converge as it was already close to the correct solution. Without this initial guess, the fully coupled problem would require timestep cuts to slowly reach the final solution. The effectiveness of this preconditioner was also tested on a two-dimensional heterogeneous condensation problem and proved to converge for one to two orders of magnitude larger CFL timesteps than the fully coupled problem.

From the work described in this chapter, we also demonstrated a convincing example of how a sequential-implicit strategy is able to better solve these complex nonlinear multiphysics problems. By dividing the problem into their different physics and subdomains, we are able to overcome the issues faced by the fully coupled approach. This further illustrates how applying sequential-implicit ideas to a complex multiphysics problem such as geothermal simulation is vital to both the understanding and design of better nonlinear solvers.

A tight coupling between the flow and thermal equations is also evident in thermal compositional reservoir simulation. In thermal compositional reservoir simulation, a more complex process is required to determine the thermodynamic conditions of a block. This process often involves solving an equation of state model to determine the number of phases present with the phase compositions. When there is a significant amount of water in the compositional mixture, similar "negative compressibility" effects can also be observed [10]. The ideas from the methods described in this chapter could be translated to these thermal compositional simulations. The most

important step would be to determine which of the mass equations would be solved in the m-SFI-P step for thermal compositional simulations. This would first require a sequential split of the thermal equations from the flow equations for compositional simulations [65, 66].

The findings from the work described in this chapter would have a direct application into real geothermal reservoir simulators where this condensation problem is faced. Specifically, this would occur in conventional geothermal systems with two-phase regions and cold water invading those regions. This is specifically important for the forecasting of the re-injection of cold water into the reservoir. This improvement in the nonlinear convergence would improve the forecasting and effectiveness of geothermal simulators.

Chapter 5

Conclusions

This thesis described various sequential-implicit methods that reduce the computational cost of subsurface simulations by improving their nonlinear convergence. This improvement in the nonlinear convergence results in an improvement in the robustness and speed of geothermal simulations, which in turn will allow geothermal simulators to be better tools for the forecasting and management of geothermal reservoirs. These sequential-implicit methods also provide a means to split the multiphysics problem into their different individual physics 'modules' to better understand the nonlinearities and coupling of the problem. The advances achieved in this dissertation can be used to develop specialized solvers, including multiscale finite-volume and linear-solver preconditioning methods.

The work in this thesis focused on the simulation of coupled two-phase single-component (water) flow and transport with thermal and phase change effects. In this multiphysics problem, the flow and thermal equations are strongly coupled through the thermodynamic properties that are highly nonlinear across phase boundaries and in two-phase regions. Therefore, splitting these two strongly coupled problems makes it very challenging for current sequential-implicit methods. A sequential-implicit scheme that can split these two different physical processes in geothermal simulation

is highly desirable. In addition, the sequential-implicit approach described here can be translated into a general sequential-implicit framework capable of dealing with other subsurface multiphysics problems, such as flow and transport, or flow and geomechanics. In the work described in Chapters 2, 3 and 4, we showed the progression of the development of these sequential-implicit methods for geothermal reservoir simulation.

In Chapter 2, we investigated two different sequential-implicit methods for geothermal simulation. Inspired by the literature on sequential-implicit methods for modeling coupled fluid flow and geomechanics [28, 30, 29], we investigated various constraints when solving the flow and thermal residual equations. The findings from the study showed that a hybrid constraint, where a fixed pressure was enforced for single-phase cells and a fixed density was enforced for two-phase cells performed the best out of the different constraint strategies. However, when comparing this constraint strategy with the fully coupled approach for complex reservoir models, the sequential-implicit strategy required a large number of outer loop iterations to converge to the solution. This made the sequential strategy undesirable, as the large number of sequential iterations diminished any of the computational gains from splitting the problem. The second sequential-implicit strategy investigated was a modified sequential fully implicit method. This was first developed for flow and transport problems by Moncorgé et al. [42], where they enriched the flow residual equations with additional transport equations. Here we enriched the flow equations by including additional thermal residual equations. We investigated various strategies to enrich the flow equations that adapted to the flow regime of the problem. From the numerical models tested, the additional cost of including the thermal residual equations could overshadow the computational gains from solving the problem sequentially. One of the main conclusions was that current sequential-implicit strategies applied in other multiphysics problems are insufficient for an efficient sequential-implicit strategy for geothermal reservoir simulation. This insufficiency is a result of the slow linear convergence rate

of sequential-implicit methods.

The core challenge associated with the sequential-implicit methods presented presented in Chapter 2 is the slow outer loop convergence. In work presented in Chapter 3, we developed a sequential-implicit Newton's method to improve the outer loop convergence of any sequential-implicit method. The sequential-implicit Newton's method presented in Chapter 3 is general and applicable to all multiphysics sequential-implicit simulations. To demonstrate the generality and effectiveness of this sequential-implicit Newton's method, the performance of this algorithm was presented for geothermal simulations and flow and mechanics simulations. The underlying idea for this algorithm is that all sequential-implicit methods could be reformulated as a fixed-point iteration that solves a system of nonlinear equations. To improve on this fixed-point iteration a Newton's method update could be used instead. This allows the update strategy to follow a quadratic convergence rate instead of the linear convergence rate of the fixed-point iteration. We demonstrated that with this sequential-implicit Newton's method, careful understanding of the constraints and variables is still required for convergence. This is in agreement with the results from Chapter 2 and sequential-implicit analysis for other multiphysics problems [28, 30, 29]. The numerical experiments tested here showed an improvement in outer loop convergence across all the multiphysics problems and test cases considered. For some specific cases where there is a strong coupling, there were up to two orders of magnitude decrease in the number of sequential iterations for the sequential-implicit Newton method. The convincing results in this chapter demonstrates both the effectiveness and generality of this method.

In Chapter 4, we presented a sequential-implicit nonlinear solver for the condensation problem in geothermal simulations. The difficulties arising from this condensation problem are a result of the strong coupling between the mass and energy conservation equations through the thermodynamic phase behavior. In particular,

for geothermal problems, this tight coupling results in an apparent "negative compressibility" for cells (control volumes) that have two fluid phases (liquid and vapor) during the time interval of interest. In this chapter, we presented a nonlinear solution strategy that overcomes the negative compressibility associated with the condensation front. We first demonstrated on a single-cell problem how the fully coupled method is affected by this negative compressibility phenomenon. Then, we described a sequential-implicit preconditioning solution strategy that overcomes these nonlinear convergence difficulties. The core principle of this preconditioner is that it enables the near-elliptic pressure solution from the flow equation to first determine where the phase front should be. Using this as an initial guess, the fully coupled problem was convergent as it was already close to the correct solution. Without this initial guess, the fully coupled problem would require many timestep cuts to slowly reach the final solution.

We also derived a criterion that can be used as a nonlinear preconditioning strategy to solve general multicell geothermal problems. The effectiveness of this preconditioner was tested on a two-dimensional heterogeneous condensation problem and proved to converge for one to two orders of magnitude larger CFL timesteps than the fully coupled problem. The results from this chapter further demonstrate how a strong understanding of the sequential-implicit method can lead to further improvements in a fully coupled method. The results present a convincing example of how a sequential-implicit method could be superior to a fully coupled method at tackling these complex multiphysics problems.

These advances in the design of sequential-implicit methods for geothermal simulation are important to the efficiency and understanding of the nonlinear issues in geothermal simulation. Now with suitable sequential-implicit methods to split the flow and thermal physics in geothermal simulations, specialized solvers could be designed to better improve the solution of the individual physical subproblem. One of

the key advancements in this work was the development of the sequential-implicit Newton solver. In this dissertation, we presented convincing numerical examples for geothermal and geomechanics simulations. Further analysis on other multiphysics problems such as flow-transport or three physics problems such as flow-thermal and mechanics would greatly further the capability of this algorithm.

Future Work

Flux Function Analysis

The main focus of this research was to investigate sequential-implicit algorithms to know how to split the flow and thermal equations in geothermal simulation. The next step would be construct nonlinear solvers that would be able to handle the nonlinear hyperbolic part of the problem. Similar analyses for efficient solution of isothermal transport problems have been done [27, 72, 34, 23, 24]. In those works they investigated the nonlinearities associated with the flux function of the transport equations. From an understanding of these nonlinearities, sophisticated nonlinear strategies have been developed [27, 72, 34] to overcome the different kinks and discontinuities in the problem. The analysis of geothermal problems could greatly improve the convergence of similar type problems. Similar discontinuities are likely be associated with the phase change related to the condensation problem.

Sequential-implicit Newton's Method For Multiscale Simulations

The sequential-implicit method for isothermal multiphase flow and transport simulation was first developed in the context of a Multiscale Finite Volume method [26]. In that context, the sequential-implicit method was utilized to separate out the elliptic

flow part of the problem to apply a multiscale solver on top of it. These multiscale methods have been developing rapidly over the past few years and often promise an order of magnitude improvement over traditional methods. However, one of the key bottlenecks in these multiscale methods is the slow sequential coupling between the flow and transport problems [43, 44, 32]. The sequential-implicit Newton's method strategy developed within this research would be a perfect candidate to improve upon the sequential strategy for flow and transport problems. This sequential-implicit Newton's method would likely overcome many of the sequential coupling issues that are seen for strong capillary forces between multiphase flow. This could provide tremendous computational gains for the entire simulation workflow of multiscale simulations.

Thermal Compositional Simulation

The focus of the work described in this thesis was on simulating the flow and transport for a single-component fluid (water) in two phases with thermal effects. This allowed us to only require the thermodynamic relationships for water, resulting in simpler analysis of the complex coupling between the flow and thermal problems. Further work on extending the m-SFI-P method for thermal compositional simulation would be extremely valuable for applications such as steam-assisted gravity segregation and thermal enhanced oil recovery methods. For these multicomponent systems, the thermodynamic relationships will require an equation of state model to fully determine the thermodynamic state and fluid properties of a cell. For multicomponent systems that contain water, similar "negative compressibility" effects will be present. The analysis described in Chapter 4 provides a solid foundation to build a similar sequential preconditioning strategy for compositional systems. Additionally, the issue of the "negative compressibility" is likely to be less severe than the pure water system due to the mixing of hydrocarbon and water [10]. An extension of the m-SFI-P method would likely require a similar sequential-implicit analysis for thermal compositional

simulations [65, 66]. Following a sequential structure for thermal compositional simulation, an m-SFI-P method may prove to be effective for thermal-compositional simulation.

Appendix A

AD-GPRS Input Files

```
#Geothermal Production Moving flash front

VERBOSE
DEBUG SILENT /

DIMENS
26 1 1
/

DEPTH
1000 /
/

COMPS
H2O /
#C1 C20 H2O /

ZI
1.00 /
# 0.001   0.001 0.998 /
```

```
ROCK
1 0  265.0 2.5e3/


THCROCK
1728 /

THCGAS
1.3824 /

THCOIL
1 /

THCWATER
48.643 /

SWOF
0.3     0       1       0
0.35    3.50128E-05     0.999964987     0
0.4     0.000560204     0.999439796     0
0.45    0.002836035     0.997163965     0
0.5     0.008963272     0.991036728     0
0.55    0.021882987     0.978117013     0
0.6     0.045376562     0.954623438     0
0.65    0.084065684     0.915934316     0
0.7     0.143412346     0.856587654     0
0.75    0.229718847     0.770281153     0
0.8     0.350127797     0.649872203     0
0.85    0.512622107     0.487377893     0
0.9     0.726024999     0.273975001     0
0.95    1       0       0
```

```
/

SGOF
0.05    0       1       0
0.1     0.000875319     0.999124681     0
0.15    0.006722454     0.993277546     0
0.2     0.021742936     0.978257064     0
0.25    0.049297994     0.950702006     0
0.3     0.091908547     0.908091453     0
0.35    0.151255208     0.848744792     0
0.4     0.228178285     0.771821715     0
0.45    0.322677777     0.677322223     0
0.5     0.433913378     0.566086622     0
0.55    0.560204475     0.439795525     0
0.6     0.699030146     0.300969854     0
0.65    0.847029166     0.152970834     0
0.7     1       0       0
/


TEMP
573.15 /


PRESSURE
90/

PORO
0.2 /

SGAS
0 /
```

```
SCOND
1.0 288.706 /

TPFACONNSN
25
0       1       156.632807      1812.944057
1       2       156.632807      1812.944057
2       3       156.632807      1812.944057
3       4       156.632807      1812.944057
4       5       156.632807      1812.944057
5       6       156.632807      1812.944057
6       7       156.632807      1812.944057
7       8       156.632807      1812.944057
8       9       156.632807      1812.944057
9       10      156.632807      1812.944057
10      11      156.632807      1812.944057
11      12      156.632807      1812.944057
12      13      156.632807      1812.944057
13      14      156.632807      1812.944057
14      15      156.632807      1812.944057
15      16      156.632807      1812.944057
16      17      156.632807      1812.944057
17      18      156.632807      1812.944057
18      19      156.632807      1812.944057
19      20      156.632807      1812.944057
20      21      156.632807      1812.944057
21      22      156.632807      1812.944057
22      23      156.632807      1812.944057
23      24      156.632807      1812.944057
24      25      156.632807      1812.944057
/
```

```
VOLUME
97.817059
97.817059
224.724576
412.839653
871.776601
1685.529459
3444.110698
6796.262020
13708.294782
27270.849844
54725.172595
109219.374661
218729.517651
437092.990778
874646.792801
1748713.475070
3498157.246875
6995395.129983
13991947.639441
27982438.264168
55966710.751799
111931112.415727
223865131.722358
447726603.985011
895457814.832298
1790909830.125583
/


NONLINEAR
```

```
FLOW 1e-4 20 0 APPL /
THERMAL 1e-4 20/
/

WELSPECS
P1 * 1 1 1000/
/

COMPDAT
P1 1 1 1 1   OPEN 1* 100 0.24384 3* Z /
/

WCONPROD
P1      OPEN    WRAT    1*      1209600 3* 0 /
/

OUTPUTVARS
PRES TEMP SAT ENTHALPY ENTHALPY_TOT DENSITY DENSITY_TOT  /
/

WELL_RATES_AT
RESERVOIR_CONDITIONS /

OUTPUT
HDF5 TIME "OUTPUT" /

TUNING
1e-7  1000  * 2 10.0 10.0 0.2 0.1 /

TSTEP
1*1 /
```

```
end
```

Figure A.1: Full AD-GPRS Input File for One-dimensional radial model input for two-phase production case

```
VERBOSE
DEBUG DEBUG /

# Include SPE10 permeability and porosity
INCLUDE
../grid.inc /

DEPTH
1000 /
/

COMPS
H2O /

ZI
1.00 /

ROCK
1 0  265.0 2.5e3/



THCWATER
48.643 /

SWOF
0.3     0       1       0
0.35    3.50128E-05     0.999964987     0
0.4     0.000560204     0.999439796     0
0.45    0.002836035     0.997163965     0
0.5     0.008963272     0.991036728     0
0.55    0.021882987     0.978117013     0
0.6     0.045376562     0.954623438     0
```

```
0.65    0.084065684     0.915934316     0
0.7     0.143412346     0.856587654     0
0.75    0.229718847     0.770281153     0
0.8     0.350127797     0.649872203     0
0.85    0.512622107     0.487377893     0
0.9     0.726024999     0.273975001     0
0.95    1       0       0
/

SGOF
0.05    0       1       0
0.1     0.000875319     0.999124681     0
0.15    0.006722454     0.993277546     0
0.2     0.021742936     0.978257064     0
0.25    0.049297994     0.950702006     0
0.3     0.091908547     0.908091453     0
0.35    0.151255208     0.848744792     0
0.4     0.228178285     0.771821715     0
0.45    0.322677777     0.677322223     0
0.5     0.433913378     0.566086622     0
0.55    0.560204475     0.439795525     0
0.6     0.699030146     0.300969854     0
0.65    0.847029166     0.152970834     0
0.7     1       0       0
/



SCOND
1.0 288.706 /

THCROCK
1728 /
```

```
THCGAS
1.3824 /

THCOIL
1 /

THCWATER
48.643 /

OUTPUTVARS
PRES SAT_NP TEMP ENTHALPY_TOT ENTHALPY CFL DENSITY_NP STATUS/
/

WELL_RATES_AT
RESERVOIR_CONDITIONS /

OUTPUT
HDF5 TIME "OUTPUT" /

TUNING
10 10000 * 5 5* 10000 /

TSTEP
50
/
end
```

Figure A.2: AD-GPRS Input File for two-dimensional SPE10 model

```
VERBOSE
DEBUG DEBUG /
/

DIMENS
4 5 6/
/
DX
 1000/
/
DY
 1000/
/
DZ
 300/
/
TEMP
20*433.15 20*553.15 20*553.15 20*553.15 20*553.15 20*553.15  /
/
PRESSURE
20*40.00 20*64.00 20*88.00 20*112.00 20 19*136.00 20*160.00 /
/
SGAS
20*0.00 20*0.15 20*0.00 20*0.00 20*0.00 20*0.00 /
/
DEPTH
20*150.00 20*450.00 20*750.00 20*1050.00 20*1350.00 20*1650.00 /
/
PERMX
20*100.00 20*200.00 20*200.00 20*200.00 20*100.00 20*100.00 /
/
PERMY
```

```
20*100.00 20*200.00 20*200.00 20*200.00 20*100.00 20*100.00 /
/
PERMZ
20*2.00 20*50.00 20*50.00 20*50.00 20*2.00 20*2.00 /
/
PORO
20*0.20 20*0.25 20*0.25 20*0.25 20*0.20 20*0.20 /
/

ROCK
1 0  265.0 2.5e3/

THCROCK
86.4
/

THCGAS
1.3824 /

THCOIL
1 /

THCWATER
48.643 /

COMPS
H2O /
ZI
1.0/
/
ROCK
68.9476 7.2519e-5 /
```

```
/
SWOF
0.3     0       1       0
0.35    3.50128E-05     0.999964987     0
0.4     0.000560204     0.999439796     0
0.45    0.002836035     0.997163965     0
0.5     0.008963272     0.991036728     0
0.55    0.021882987     0.978117013     0
0.6     0.045376562     0.954623438     0
0.65    0.084065684     0.915934316     0
0.7     0.143412346     0.856587654     0
0.75    0.229718847     0.770281153     0
0.8     0.350127797     0.649872203     0
0.85    0.512622107     0.487377893     0
0.9     0.726024999     0.273975001     0
0.95    1       0       0
/
SGOF
0.05    0       1       0
0.1     0.000875319     0.999124681     0
0.15    0.006722454     0.993277546     0
0.2     0.021742936     0.978257064     0
0.25    0.049297994     0.950702006     0
0.3     0.091908547     0.908091453     0
0.35    0.151255208     0.848744792     0
0.4     0.228178285     0.771821715     0
0.45    0.322677777     0.677322223     0
0.5     0.433913378     0.566086622     0
0.55    0.560204475     0.439795525     0
0.6     0.699030146     0.300969854     0
0.65    0.847029166     0.152970834     0
0.7     1       0       0
```

```
/
SCOND
1.0135 288.89/
/
MINPV
1e-11
/
WELSPECS
PROD1 * 1 1 */
/
COMPDAT
PROD1 1 1 4 4 OPEN * 500.0 4* Z /
/
WELL_RATES_AT
RESERVOIR_CONDITIONS /

TUNING
5e-4 150 0  2.5.
/

OUTPUTVARS
PRES SAT_NP TEMP ENTHALPY_TOT ENTHALPY CFL /
/

OUTPUT
HDF5 TIME "OUTPUT"
/

WCONPROD
PROD1 OPEN      WRAT    1*   8460000 3* 0 /
/
```

```
TSTEP
1*3650 /


end
```

Figure A.3: AD-GPRS Input File for Three-dimensional model with two-phase gravity drainage

```
VERBOSE
DEBUG DEBUG /
DIMENS
11 1 1/

TEMP
355.00 10*455.15 /

PRESSURE
90.00 10*10.00 /

SGAS
 0.00 10*0.90 /

TPFACONNSN
10
0 1 5000.000000 5000.000000
1 2 2500.000000 2500.000000
2 3 2500.000000 2500.000000
3 4 2500.000000 2500.000000
4 5 2500.000000 2500.000000
5 6 2500.000000 2500.000000
6 7 2500.000000 2500.000000
7 8 2500.000000 2500.000000
8 9 2500.000000 2500.000000
9 10 2500.000000 2500.000000
/
VOLUME
1000000000000000.000000
10000.000000
10000.000000
10000.000000
```

```
10000.000000
10000.000000
10000.000000
10000.000000
10000.000000
10000.000000
10000.000000
/

DEPTH
1000 /
/

COMPS
H2O /

ZI
1.00 /

ROCK
1 0  265.0 2.5e3/


THCWATER
48.643 /

SWOF
0.3     0       1       0
0.35    3.50128E-05     0.999964987     0
0.4     0.000560204     0.999439796     0
0.45    0.002836035     0.997163965     0
0.5     0.008963272     0.991036728     0
```

```
0.55    0.021882987     0.978117013     0
0.6     0.045376562     0.954623438     0
0.65    0.084065684     0.915934316     0
0.7     0.143412346     0.856587654     0
0.75    0.229718847     0.770281153     0
0.8     0.350127797     0.649872203     0
0.85    0.512622107     0.487377893     0
0.9     0.726024999     0.273975001     0
0.95    1       0       0
/

SGOF
0.05    0       1       0
0.1     0.000875319     0.999124681     0
0.15    0.006722454     0.993277546     0
0.2     0.021742936     0.978257064     0
0.25    0.049297994     0.950702006     0
0.3     0.091908547     0.908091453     0
0.35    0.151255208     0.848744792     0
0.4     0.228178285     0.771821715     0
0.45    0.322677777     0.677322223     0
0.5     0.433913378     0.566086622     0
0.55    0.560204475     0.439795525     0
0.6     0.699030146     0.300969854     0
0.65    0.847029166     0.152970834     0
0.7     1       0       0
/

PORO
0.2 /

SCOND
```

```
1.0 288.706 /

THCROCK
1728 /

THCGAS
1.3824 /

THCOIL
1 /

THCWATER
48.643 /

OUTPUTVARS
PRES SAT_NP TEMP ENTHALPY_TOT ENTHALPY CFL /
/

WELL_RATES_AT
RESERVOIR_CONDITIONS /
```

Figure A.4: AD-GPRS Input File for one dimensional model for negative compressibility

Bibliography

[1] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[2] G. Axelsson, V. Stefánsson, and Y. Xu. Sustainable management of geothermal resources. In *Proceedings of the International Geothermal Conference*, pages 40–48, 2003.

[3] K. Aziz and A. Settari. *Petroleum reservoir simulation*. Chapman & Hall, 1979.

[4] R. Barrett, M.W. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*, volume 43. Siam, 1994.

[5] Jakub Wiktor Both, Kundan Kumar, Jan Martin Nordbotten, and Florin Adrian Radu. Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media. *arXiv preprint arXiv:1805.04211*, 2018.

[6] J. Burnell, E. Clearwater, A. Croucher, W. Kissling, J.P. O'Sullivan, M.J. O'Sullivan, and A. Yeh. Future directions in geothermal modelling. In *Proceedings (electronic) 34rd New Zealand Geothermal Workshop*, pages 19–21, 2012.

[7] N. Castelletto, J.A. White, and H.A. Tchelepi. Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics.

International Journal for Numerical and Analytical Methods in Geomechanics, 39(14):1593–1618, 2015.

[8] M.A. Christie and M.J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2001.

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

[10] K.H. Coats. Reservoir simulation: A general model formulation and associated physical/numerical sources of instability. *Boundary and Interior Layers-Computational and Asymptotic Methods*, pages 62–76, 1980.

[11] O. Coussy. *Poromechanics*. John Wiley & Sons, 2004.

[12] A. Croucher, J.P. O'Sullivan, A. Yeh, and M.J. O'Sullivan. Benchmarking and experiments with waiwera, a new geothermal simulator. In *43rd Workshop on Geothermal Reservoir Engineering*, 2018.

[13] V. Dolean, M.J. Gander, W. Kheriji, F. Kwok, and R. Masson. Nonlinear preconditioning: How to use a nonlinear schwarz method to precondition newton's method. *SIAM Journal on Scientific Computing*, 38(6):A3357–A3380, 2016.

[14] R.W. Falta, K. Pruess, I. Javandel, and P.A. Witherspoon. Numerical modeling of steam injection for the removal of nonaqueous phase liquids from the subsurface: 1. numerical formulation. *Water Resources Research*, 28(2):433–449, 1992.

[15] C.R. Faust and J.W. Mercer. Geothermal reservoir simulation: 2. numerical solution techniques for liquid-and vapor-dominated hydrothermal systems. *Water Resources Research*, 15(1):31–46, 1979.

[16] Martin J Gander. Optimized Schwarz methods. *SIAM Journal on Numerical Analysis*, 44(2):699–731, 2006.

[17] Martin J Gander. Schwarz methods over the course of time. *Electron. Trans. Numer. Anal*, 31(5):228–255, 2008.

[18] T.T. Garipov, P. Tomin, R. Rin, D.V. Voskov, and H.A. Tchelepi. Unified thermo-compositional-mechanical framework for reservoir simulation. *Computational Geosciences*, pages 1–19, 2018.

[19] T.T. Garipov, D.V. Voskov, and H.A. Tchelepi. Rigorous coupling of geomechanics and thermal-compositional flow for sagd and es-sagd operations. In *SPE Canada Heavy Oil Technical Conference*. Society of Petroleum Engineers, 2015.

[20] T.T. Garipov, J.A. White, A. Lapene, and H.A. Tchelepi. Thermo-hydro-mechanical model for source rock thermal maturation. In *50th US Rock Mechanics/Geomechanics Symposium*. American Rock Mechanics Association, 2016.

[21] J. Gudbjerg, O. Trötschler, A. Färber, T.O. Sonnenborg, and K.H. Jensen. On spurious water flow during numerical simulation of steam injection into water-saturated soil. *Journal of Contaminant Hydrology*, 75(3):297–318, 2004.

[22] H. Hajibeygi and H.A. Tchelepi. Compositional multiscale finite-volume formulation. *SPE Journal*, 19(02):316–326, 2014.

[23] F.P. Hamon, B.T. Mallison, and H.A. Tchelepi. Implicit hybrid upwind scheme for coupled multiphase flow and transport with buoyancy. *Computer Methods in Applied Mechanics and Engineering*, 311:599–624, 2016.

[24] F.P. Hamon and H.A. Tchelepi. Analysis of hybrid upwinding for fully-implicit simulation of three-phase flow with gravity. *SIAM Journal on Numerical Analysis*, 54(3):1682–1712, 2016.

[25] P. Jenny, S.H. Lee, and H.A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187(1):47–67, 2003.

[26] P. Jenny, S.H. Lee, and H.A. Tchelepi. Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media. *Journal of Computational Physics*, 217(2):627–641, 2006.

[27] P. Jenny, H.A. Tchelepi, and S.H. Lee. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with S-shaped flux functions. *Journal of Computational Physics*, 228(20):7497–7512, 2009.

[28] J. Kim, H.A. Tchelepi, and R. Juanes. Stability, accuracy and efficiency of sequential methods for coupled flow and geomechanics. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009.

[29] J. Kim, H.A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2094–2116, 2011.

[30] J. Kim, H.A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Computer Methods in Applied Mechanics and Engineering*, 200(13):1591–1606, 2011.

[31] S. Klevtsov, N. Castelletto, J.A. White, and H.A. Tchelepi. Block-preconditioned krylov methods for coupled multiphase reservoir flow and geomechanics. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[32] A. Kozlova, Z. Li, J.R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, and S.H. Lee. A real-field multiscale black-oil reservoir simulator. *SPE Journal*, 21(06):2–049, 2016.

[33] S.H. Lee, C. Wolfsteiner, and H.A. Tchelepi. Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. *Computational Geosciences*, 12(3):351–366, 2008.

[34] B. Li and H.A. Tchelepi. Nonlinear analysis of multiphase transport in porous media in the presence of viscous, buoyancy, and capillary forces. *Journal of Computational Physics*, 297:104–131, 2015.

[35] Xiaoye S Li and James W Demmel. Superlu_dist: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Transactions on Mathematical Software (TOMS)*, 29(2):110–140, 2003.

[36] X.S. Li. An overview of superlu: Algorithms, implementation, and user interface. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):302–325, 2005.

[37] L. Liu and D.E. Keyes. Field-split preconditioned inexact newton algorithms. *SIAM Journal on Scientific Computing*, 37(3):A1388–A1409, 2015.

[38] L. Magnúsdóttir. *Fracture Characterization in Geothermal Reservoirs Using Time-lapse Electric Potential Data*. PhD thesis, Stanford University, 2013.

[39] J. Mandel. Consolidation des sols (étude mathématique). *Geotechnique*, 3(7):287–299, 1953.

[40] A. Mikelić and M.F. Wheeler. Convergence of iterative coupling for coupled flow and geomechanics. *Computational Geosciences*, 17(3):455–461, 2013.

[41] A. Moncorgé and H.A. Tchelepi. Negative compressibilities for steam/water flow in porous media. Unpublished Manuscript, 2017.

[42] A. Moncorgé, H.A. Tchelepi, and P. Jenny. Modified sequential fully implicit scheme for compositional flow simulation. *Journal of Computational Physics*, 337:98–115, 2017.

[43] A. Moncorgé, H.A. Tchelepi, and P. Jenny. Sequential fully implicit formulation for compositional simulation using natural variables. *Journal of Computational Physics*, 371:690–711, 2018.

[44] O. Møyner and K.A. Lie. A multiscale restriction-smoothed basis method for compressible black-oil models. *SPE Journal*, 21(06):2–079, 2016.

[45] O. Møyner and H.A. Tchelepi. A multiscale restriction-smoothed basis method for compositional models. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[46] D.J. Noy, S. Holloway, R.A. Chadwick, J.D.O. Williams, S.A. Hannis, and R.W. Lahann. Modelling large-scale carbon dioxide injection into the bunter sandstone in the UK southern north sea. *International Journal of Greenhouse Gas Control*, 9:220–233, 2012.

[47] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.

[48] M. J. O'Sullivan, K. Pruess, and M.J. Lippmann. State of the art of geothermal reservoir simulation. *Geothermics*, 30(4):395–429, 2001.

[49] J.P. O'Sullivan, A. Croucher, A. Yeh, and M.J. O'Sullivan. Further improvements in the convergence of tough2 simulations. In *11th World Congress on Computational Mechanics, Barcelona, Spain*, 2014.

[50] C.G. Petra, O. Schenk, and M. Anitescu. Real-time stochastic optimization of complex energy systems on high-performance computers. *Computing in Science & Engineering*, 16(5):32–42, 2014.

[51] C.G. Petra, O. Schenk, M. Lubin, and K. Gärtner. An augmented incomplete factorization approach for computing the schur complement in stochastic optimization. *SIAM Journal on Scientific Computing*, 36(2):C139–C162, 2014.

[52] H.S. Price and K.H. Coats. Direct methods in reservoir simulation. *Society of Petroleum Engineers Journal*, 14(03):295–308, 1974.

[53] K. Pruess, C. Calore, R. Celati, and Y.S. Wu. An analytical solution for heat transfer at a boiling front moving through a porous medium. *International Journal of Heat and Mass Transfer*, 30(12):2595–2602, 1987.

[54] Karsten Pruess, CM Oldenburg, and GJ Moridis. *TOUGH2 user's guide version 2*. Lawrence Berkeley National Laboratory, 1999.

[55] R. Rin, P. Tomin, T.T. Garipov, D.V. Voskov, and H.A. Tchelepi. General implicit coupling framework for multi-physics problems. In *SPE Reservoir Simulation Conference*. Society of Petroleum Engineers, 2017.

[56] J. Rutqvist, Y.S. Wu, C.F. Tsang, and G. Bodvarsson. A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock. *International Journal of Rock Mechanics and Mining Sciences*, 39(4):429–442, 2002.

[57] Y. Saad and M.H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[58] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2nd edition, 2003.

[59] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Generation Computer Systems*, 20(3):475–487, 2004.

[60] O. Schenk and K. Gärtner. On fast factorization pivoting methods for sparse symmetric indefinite systems. *Electronic Transactions on Numerical Analysis*, 23(1):158–179, 2006.

[61] Amik St-Cyr, Martin J Gander, and Stephen J Thomas. Optimized multiplicative, additive, and restricted additive Schwarz preconditioning. *SIAM Journal on Scientific Computing*, 29(6):2402–2425, 2007.

[62] (SGP) Stanford Geothermal Program. Proceedings of the special panel on geothermal model study, report SGP-TR-42. *Energy Resources Engineering, Stanford University*, 1980.

[63] J.L. Sullivan, C.E. Clark, J. Han, and M. Wang. Life-cycle analysis results of geothermal systems in comparison to other power systems. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2010.

[64] Alex Toth and CT Kelley. Convergence analysis for Anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.

[65] J.A. Trangenstein. Analysis of a model and sequential numerical method for thermal reservoir simulation. In *ECMOR I-1st European Conference on the Mathematics of Oil Recovery*, 1989.

[66] D.E.A. van Odyck, J.B. Bell, F. Monmont, and N. Nikiforakis. The mathematical structure of multiphase thermal models of flow in porous media. In *Proceedings*

of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 465, pages 523–549. The Royal Society, 2009.

[67] R.S. Varga. On diagonal dominance arguments for bounding $||A^{-1}||_\infty$. *Linear Algebra and its Applications*, 14(3):211–217, 1976.

[68] D.V. Voskov and H.A. Tchelepi. Comparison of nonlinear formulations for two-phase multi-component EoS based simulation. *Journal of Petroleum Science and Engineering*, 82:101–111, 2012.

[69] D.V. Voskov, R. Zaydullin, and A. Lucia. Heavy oil recovery efficiency using sagd, sagd with propane co-injection and strip-sagd. *Computers & Chemical Engineering*, 88:115–125, 2016.

[70] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

[71] J.R. Wallis. Incomplete gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1983.

[72] X. Wang and H.A. Tchelepi. Trust-region based solver for nonlinear transport in heterogeneous porous media. *Journal of Computational Physics*, 253:114–137, 2013.

[73] Y. Wang. A stability criterion for the negative compressibility problem in geothermal simulation and discrete modeling of failure in oil shale pyrolysis process. Master's thesis, Stanford University, 2015.

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

[75] J.A. White, N. Castelletto, and H.A. Tchelepi. Block-partitioned solvers for coupled poromechanics: A unified framework. *Computer Methods in Applied Mechanics and Engineering*, 303:55–74, 2016.

[76] Z.Y. Wong, R.N. Horne, and D.V. Voskov. A geothermal reservoir simulator in AD-GPRS. In *Proceedings World Geothermal Congress*, 2015.

[77] Z.Y. Wong, R.N. Horne, and D.V. Voskov. Comparison of nonlinear formulations for geothermal reservoir simulations. In *41st Workshop on Geothermal Reservoir Engineering*, 2016.

[78] Z.Y. Wong, R. Rin, H.A. Tchelepi, and R.N. Horne. Comparison of a fully implicit and sequential implicit formulation for geothermal reservoir simulations. In *42nd Workshop on Geothermal Reservoir Engineering*, 2017.

[79] R.M. Younis. *Modern advances in software and solution algorithms for reservoir simulation*. PhD thesis, Stanford University, 2011.

[80] R. Zaydullin, D.V. Voskov, S.C. James, H. Henley, and A. Lucia. Fully compositional and thermal reservoir simulation. *Computers & Chemical Engineering*, 63:51–65, 2014.

[81] R. Zaydullin, D.V. Voskov, and H.A. Tchelepi. Formulation and solution of compositional displacements in tie-simplex space. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2013.

[82] Y. Zhou. *Multistage preconditioner for well groups and automatic differentiation for next generation GPRS*. PhD thesis, Master's thesis, Stanford University, 2009.

[83] Y. Zhou. *Parallel general-purpose reservoir simulation with coupled reservoir models and multisegment wells*. PhD thesis, Stanford University, 2012.

[84] Y. Zhou, Y. Jiang, and H.A. Tchelepi. A scalable multistage linear solver for reservoir models with multisegment wells. *Computational Geosciences*, 17(2):197–216, 2013.

[85] Y. Zhou, H.A. Tchelepi, and B.T. Mallison. Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2011.

[86] O.C Zienkiewicz and R.L. Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 2005.