A GEOTHERMAL RESERVOIR SIMULATOR IN AD-GPRS

A REPORT
SUBMITTED TO THE DEPARTMENT OF ENERGY
RESOURCES ENGINEERING
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTERS OF SCIENCE

Zhi Yang Wong
June 2015

© Copyright by Zhi Yang Wong 2015
All Rights Reserved

I certify that I have read this report and that, in my opinion, it is fully adequate in scope and quality as a report for the degree of Masters of Science in Petroleum Engineering.

(Roland Horne) Principal Adviser

Abstract

The AD-GPRS framework was modified to simulate geothermal reservoirs. AD-GPRS (automatic differentiation general purpose research simulator) is a computational framework that allows for fully compositional and thermal reservoir simulation. This study looked specifically at the geothermal single-component, two-phase case. AD-GPRS is based on an automatic differentiation library that allows for a flexible treatment of the variable formulation and the nonlinear terms within the formulation. This allows a geothermal application functionality to be integrated within this framework.

Two nonlinear formulations were developed and implemented using the AD-GPRS framework: a pressure, temperature and saturation formulation and a pressure, enthalpy formulation. To verify the results obtained from geothermal AD-GPRS, comparisons with other models were carried out. The geothermal AD-GPRS results were compared to analytical and semianalytical models to verify the solutions obtained from the simulator. For situations where an analytical solution was unavailable, TOUGH2, a commonly used reservoir simulator was used for comparison. AD-GPRS had very good agreement with the results obtained from the other methods.

Once the solutions were verified for these test cases, a comparison of the nonlinear behavior between the two formulations was carried out. In particular the behavior for the two formulations for an unstable physical phenomena of cold water injecting into hot steam known as "negative compressibility" was investigated. From these comparisons, it was found that the pressure, temperature and saturation formulation performed better than the pressure, enthalpy formulation.

Acknowledgments

First of all, I would like to thank God for all the blessings he has provided for me in my life and especially this opportunity to study at Stanford.

I am deeply indebted my advisor, Professor Roland Horne for his guidance, encouragement and support throughout my program. I am immensely grateful to Dr. Denis Voskov for his guidance and deep understanding with the technical details within my research. I also would like to thank the Stanford Geothermal Program group for their friendship and research advice. I am grateful also to the SUPRI-B research group for their inspiring weekly meetings.

I am also grateful for my family and for their support and unconditional care while I study at Stanford. I am immensely grateful for Eunice Chow for the love and support that she gives me.

I am thankful for all my friends and my office mates here at Stanford. My many late nights at the office would have been extremely difficult without your laughter and camaraderie. I am appreciative for the Stanford IVGrad group for their friendship and community during my time here at Stanford.

Finally, I would also like to thank the GTO-CCS and the Stanford Geothermal Program for their financial support of this research.

Nomenclature

μ_k	Viscosity of the phase k
μ_s	Dynamic viscosity of steam
μ_w	Dynamic viscosity of water
ϕ	Porosity of the rock
ρ	Density of fluid
ρ_s	Saturated steam density
ρ_w	Saturated water density
f	Degrees of freedom
g	Gravitational constant
h	Fluid enthalpy
H_k	Internal energy of phase k
h_s	Saturated steam enthalpy
h_w	Saturated water enthalpy
K	Total conductivity of the fluid and rock
k_{rk}	Relative permeability
N_c	Number of components

N_p	Number of phases
p	Pressure
p_{sat}	Saturation pressure
Q	Source/sink term
S_s	Steam saturation
S_w	Water saturation
T	Temperature
T_k^i	Phase transmissibility of interface i of phase k
u_k	Velocity of the phase k
z	Coordinate direction of gravity
m	Meters
ph	Pressure-enthalpy
pTS	Pressure-temperature-saturation
r	Position radially from the wellbore
t	Time

Contents

Abstract **iv**

Acknowledgments **v**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 AD-GPRS 3

1.3 Objectives 5

2 Mathematical Formulation **6**

2.1 Governing Equations 6

2.1.1 Conservation Equations 6

2.1.2 Darcy's Law 8

2.2 Solution Procedure 8

2.3 Primary Variables 12

2.4 Pressure-Temperature-Saturation Formulation 13

2.4.1 Phase Change Algorithm 14

2.5 Pressure-Enthalpy Formulation 15

2.5.1 Phase Behavior 16

2.5.2 Basic Phase Change Algorithm 17

2.5.3 Phase Behavior Update with Chop Algorithm 18

2.6 Thermodynamic Relations 21

2.6.1 Thermodynamic Relations for ph Formulation 21

2.6.2 Thermodynamic Relations for pTS Formulation 23

2.7 Concluding Remarks 25

3 AD-GPRS **26**

3.1 Numerical Framework Based on ADETL 27

3.1.1 ADETL Implementation 27

3.1.2 Inverse Method 28

3.1.3 ph Formulation Inverse Method 30

3.1.4 pTS Formulation Inverse Method 31

3.2 Programming Structure of AD-GPRS 31

3.2.1 NonlinearFormulation Object 32

3.2.2 Fluid 35

3.2.3 Other Objects and Implementation details 36

3.3 Concluding Remarks 36

4 Numerical Verification **37**

4.1 Single-phase Injection 37

4.1.1 Problem Specification 37

4.1.2 Results 38

4.2 Production 40

4.2.1 Results 41

4.3 Two-phase Boiling with Gravity Drainage 46

4.3.1 Results 47

4.4 Concluding Remarks 50

5 Comparison of Formulations **51**

5.1 Single Block Model 52

5.1.1 Production 54

5.1.2 Injection 55

5.2 One-dimensional Radial Model 57

5.2.1 Production 57

5.2.2 Injection 59

5.3 Concluding Remarks 69

6 Conclusions **70**

A Additional Plots **72**

List of Tables

2.1 pTS formulation status table for liquid water and steam 13
2.2 ph formulation status table for liquid water and steam 16

4.1 Model parameters for Test Case 1 38
4.2 Model parameters for Test Case 2 40
4.3 Model parameters for Test Case 3 47

5.1 Initial conditions for the different single block models. 53
5.2 Maximum time step, CFL, number of Newton Iterations and the solution after the corresponding time step for the two formulation in a production scenario (Here the superscript $n+1$ on p, S_s represents the pressure and steam saturation after one time step size of Δt) 54
5.3 Maximum time step, CFL, Newton iterations and the solution after the corresponding time step for the two formulations when injecting cold water (Here the superscript $n+1$ on p, S_s represents the pressure and steam saturation after one time step size of Δt) 55
5.4 Time step and Newton iteration results for the production from a liquid water reservoir 58

List of Figures

3.1 Structure of Nonlinear Formulation with respect to entire simulator including new inherited geothermal classes 33
3.2 Structure of Fluid with respect to entier simulator including new inherited geothermal classes 35

4.1 Temperature distribution at time = 10^9 seconds (left) and the temperature distribution corresponding to the block with a block center of 37.5m (right) over the simulated time period 39
4.2 Results from the 1980 Stanford Code Comparison Study Temperature distribution at time = 10^9 seconds (left) and the temperature distribution corresponding to the block with a block center of 37.5m (right) over the simulated time period 39
4.3 Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m) (Case 2A) 42
4.4 Pressure (left) and water saturation (right) plotted with t/r^2 for the pure two-phase case (Case 2B) 43
4.5 Results from the 1980 Stanford Code Comparison Study, Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m) 44
4.6 Pressure (left) and Water saturation (right) plotted with t/r^2 for the moving flash front case (Case 2C) 45
4.7 Results from the 1980 Stanford Code Comparison Study, Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m)(Case 2C) 46

4.8 Pressure profiles over time for three different blocks at different depths in the models (Case 3) 48
4.9 Water saturation profiles over time for three different blocks at different depths in the models (Case 3) 49
4.10 Results from the 1980 Stanford Code Comparison Study, Pressure profiles over time for three different blocks at different depths in the models (Case 3) 49
4.11 Results from the 1980 Stanford Code Comparison Study, Saturation profiles over time for three different blocks at different depths in the models (Case 3) 50

5.1 Pressure plotted against time (left) and saturation plotted against time (right) for the block at 0.5m away from the wellbore for production from pure liquid water reservoir for the pTS formulation and ph formulation based on the time stepping scheme described in 5.2.1. . . . 58
5.2 Pressure plotted against time (left) and saturation plotted against time (right) for the block at 0.5m away from the wellbore for production from pure liquid water reservoir for thes pTS formulation and ph formulation with both the same time stepping scheme 59
5.3 Water Saturation (top) and temperature (bottom) solution plotted radially at the time after 500 days as an example of a type A and type B solution 61
5.4 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 1.2 64
5.5 Pressure (top) Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 2 65

5.6 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 5 66

5.7 Pressure (top), Water Saturation (middle) and Temperature (Bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 10 67

5.8 Pressure (Top), Water Saturation (middle) and Temperature (bottom) solution plotted against time for the block at 0.5 m radius (left), the solution, plotted against the radius of the block at the final simulation time of 500 days (right). pTS formulation (blue line) and ph formulation (red dots) at initial time steps 10^{-4} time step multiplier of 10 and a maximum time step of 0.1 days. 68

A.1 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 1.2 73

A.2 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 2 74

A.3 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 5 75

A.4 Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 10 76

Chapter 1

Introduction

1.1 Background and Motivation

Numerical simulation and modeling of geothermal reservoirs is an important aspect in the forecasting and management of geothermal resources (O'Sullivan et al., 2001). Reservoir models are useful tools for predicting the performance of a particular geothermal field and also to assess the outcomes of various production scenarios (Axelsson et al., 2003). With the increase in computing power, there has been an increase in demand for models with increasingly complex geometries, (O'Sullivan et al., 2013b; Burnell et al., 2012), nonlinear physics (O'Sullivan et al., 2013a) and larger models (Yeh et al., 2011).

Due to these added complexities, often a simulator can experience numerical issues when finding a solution to a problem. For instance, Magnúsdóttir (2013) experienced difficulty when using a general purpose simulator TOUGH2 (Pruess et al., 1999), to solve a discrete fracture model. This was thought to be due to the complexity of the numerical grid, resulting in a domain error. Another problem observed in TOUGH2 was seen by Noy et al. (2012), 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 present difficulty to reservoir simulations, due to either inaccurate Jacobians or ill-conditioned Jacobians due the highly nonlinear nature of

the governing equations.

With the growth in computational power, not only are more complicated models being run but in some cases, ensembles of models are being run. These ensembles of models are needed for uncertainty evaluation and parameter estimation in the geothermal industry (Burnell et al., 2012). These inverse modeling techniques often need a large number of forward models with perturbations to different model parameters. Due to the large number of forward model runs, these forward models have to be robust and capable to be completed within a reasonable computational time frame. However under certain model conditions, due to convergence issues, small step sizes in time have to be taken and these forward simulations can become too expensive computationally.

Previous work done by O'Sullivan et al. (2014) looked at two specific convergence cases with TOUGH2 (Pruess et al., 1999). The first convergence issue addressed involved stalled simulations due to a moving unsaturated zone. It was found that the simulations stalled at a relatively small value ($\approx 10^6$ secs) prior to it reaching the total simulation time, thus not completing in a reasonable time frame. This issue was found to be due to the discontinuity in phase density between neighboring blocks. In particular, situations where the flow direction of a certain phase is from a block where the phase is present to where one where it is not, at the location where the phase is not present, a large discontinuity is created. To solve this, the phase density was calculated regardless of whether a phase was present in the block, this allowed the phase density to be more continuous across a block interface. This resulted in a significant increase in the performance of the problematic natural state simulations.

The other convergence issue examined by O'Sullivan et al. (2014) was a time step size limitation when approaching large simulation times. In natural state geothermal simulations, the standard way of finding the steady state solution is to run the simulation for a long period of time ($\approx 10^{16}$ seconds). Thus when the system approaches a steady-state, each consecutive update in time only requires a very small update. These small updates reach close to machine precision and results in convergence issues. To tackle this, an additional criterion was implemented to test the relative change in the primary variables. This addition resulted in a significant improvement

in the time step size at solutions close to the steady-state solution. However, even with these two improvements to the convergence in geothermal simulations, there are still time step limitations due to phase transitions.

Investigations in the nonlinear and linear solver in a full reservoir simulator can be difficult. Reservoir simulators currently used in the industry tend to be very complex and difficult to modify. Most existing geothermal modeling tools were developed over 25 years ago and do not take advantage of modern computational and software architecture advances (Burnell et al., 2015). This makes it difficult to implement or investigate specific convergence problems or to add additional nonlinear physics to the problem. Even if the problem is understood and changes are needed to be made, this can also be a difficult task.

In particular, one of the most complicated computations in reservoir simulation is in the construction of the Jacobian used by the nonlinear solver. The Jacobian matrix contains the derivatives of all the residual equations with respect to each independent variable in the nonlinear formulation. These residual equations and independent variables can change depending on the type of nonlinear formulation used or the physics that is modeled. Currently, most codes either employ a manual implementation of the derivatives (GeoQuest, 2001) or compute a numerical approximation of the derivatives (Pruess et al., 1999). Although a numerical Jacobian would allow for greater flexibility in the computation, this is less accurate and could result in instabilities when solving the nonlinear equations (Vanden and Orkwis, 1996). However, a manual implementation would be significantly harder to implement and is tedious when implementing additional physics or if changes to the formulation is done. These hurdles can make it difficult to implement or investigate the effects of different formulations or solution strategies for a general purpose simulator.

1.2 AD-GPRS

One method of overcoming this hurdle is to implement an automatic-differentiation library that forms the analytical Jacobian automatically based on the residual formulation. Automatic-Differentiation General Purpose Research Simulator (AD-GPRS)

(Voskov et al., 2012) is a reservoir modeling platform developed and sustained by the SUPRI-B research group in the Energy Resources Engineering Department at Stanford University. The nonlinear framework of AD-GPRS is built on the Automatic Differentiation Expression Templates Library. This library (ADETL) was developed initially by Younis (2011) and further extended by Zhou (2012).

AD-GPRS allows the flexible treatment of all nonlinear physics or formulations through the automatic differentiation framework in the simulator (Voskov et al., 2009; Voskov and Tchelepi, 2012; Iranshahr et al., 2010; Zaydullin et al., 2014). Other AD-GPRS options include flexible Multi-Point Flux Approximation discretization and general Adaptive Implicit Method (Zhou et al., 2011), OpenMP and Multi-GPU parallel implementation (Zhou, 2012; Tchelepi and Zhou, 2013), advanced thermal-compositional formulation, fully-coupled geomechanics for fractured media (Garipov et al., 2012) and adjoint-based optimization (Kourounis et al., 2010; Volkov and Voskov, 2013). A geothermal reservoir simulator built within this framework would allow these unique features to be implemented for geothermal applications.

Of these capabilities described, the flexible treatment of all nonlinear physics was of most interest to this work. Voskov and Tchelepi (2012) conducted a comparison between a range of widely used nonlinear formulations for general-purpose compositional reservoir simulation. It was found in that study that a natural variable formulation approach performed the best for isothermal compositional oil and gas simulations. It was found that for cases where a large number of phase changes occurred, the formulations with the saturation as the primary variable would provide the best guidance to the nonlinear solver.

Similar to the study done by Voskov and Tchelepi (2012), this work attempted to compare two main nonlinear formulations for geothermal reservoir simulation. The two nonlinear formulations that were investigated were a natural variable approach involving using pressure, temperature and saturation as the primary variables and a persistent variable type approach where pressure and enthalpy were the primary variables. The focus of this comparison was on the nonlinear challenge of the phase changes in geothermal simulations. These phase changes present a great difficulty for the nonlinear solver due to the large difference and discontinuity in thermodynamic

parameters between liquid water and steam.

The main difference between these two formulations is that in the pressure-enthalpy formulation the variables remain independent through phase transitions and are constant both in single-phase and two-phase regions. However for a natural variable approach with pressure and temperature, a switching in the primary variables is required. Also, the Jacobian and nonlinear convergence for these two formulations would be different due to the different choices in primary variables. A pressure-enthalpy type approach has been implemented in multiphase simulators in the past (Pritchett, 1975; Faust and Mercer, 1979; Pruess et al., 1979). One of the drawbacks of these approaches is that thermodynamic properties are often formulated in terms of the natural variables and thus calculating these can be difficult and require iterative methods (Pruess et al., 1999). The advantage of the other approach using pressure, temperature and saturation is that it would be easier to calculate the properties using these primary variables. However this requires a switching of primary variables between different single- and two-phase conditions. This study aimed to look at these two formulations and to examine and understand how the choice of the primary variables affect the nonlinear convergence.

1.3 Objectives

The objective of this work was to implement a geothermal reservoir simulator within the AD-GPRS framework such that other unique capabilities within AD-GPRS could be investigated in a geothermal context. To verify this implementation, a comparison study was conducted using test cases based on the 1980 Code Comparison Study (Stanford Geothermal Program, 1980). Once the results were verified, the pressure, temperature and saturation formulation was compared with the pressure and enthalpy formulation. This comparison was quantified by examining the behavior of the time step size and the associated convergence behavior under different flow conditions.

Chapter 2

Mathematical Formulation

This chapter describes the mathematical formulation for the geothermal problem. It begins by introducing the governing equations for geothermal reservoirs and an overview of the solution method of these governing equations. The solution method follows the typical reservoir simulation approach. An important aspect of the reservoir simulation is the nonlinear formulation and the choice of primary variables. A description of the two primary variable approaches is shown; a pressure, temperature and saturation formulation and a pressure enthalpy formulation. For comparisons between these two formulations to be made, it is crucial that the thermodynamics and phase equilibrium methods are consistent, thus the chapter will conclude with descriptions of the thermodynamic and phase behavior algorithms.

2.1 Governing Equations

2.1.1 Conservation Equations

For this study, only the transport of pure water in two-phase flow was considered. The mass and energy conservation equations for this problem are:

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

and:

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

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 conductivity of the fluid and rock.
- The subscripts k represents the phase of the fluid, R the rock and p the number of phases.

ρ_k, H_k of each phase will depend on the phase state of the fluid. At single-phase conditions, ρ_k and h_k are both functions of pressure and temperature, or any other two independent intensive thermodynamic parameters. However, at two-phase conditions, ρ_k and h_k will only depend on pressure since now pressure and temperature are dependent on each other($p = p_{sat}(T)$). In addition to these two conservation equations, the saturation constraint must be satisfied, that is, the sum of all the phase saturations is 1.

$$\sum_{k=1}^{p} S_k = 1 \quad (2.3)$$

2.1.2 Darcy's Law

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

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

where:

- u_k is the superficial 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 Solution Procedure

The finite volume method and with a fully implicit method was implemented to discretize the conservation of mass and energy equations (Equation 2.1 and 2.2) in space and time. This gives the discrete form of the conservation equations for the interface i:

$$R_c = \frac{\partial}{\partial t}V\left(\phi\sum_k^p \rho_k S_k\right) - \sum_i\left[\sum_k^p\left(\rho_k^i T_k^i \Delta\Phi_k^i\right)\right] - Q_m = 0 \tag{2.5}$$

and

$$R_e = \frac{\partial}{\partial t}\left[(1-\phi)U_R\rho_R + \phi\sum_k^p S_k\rho_k U_k\right] - \sum_i\left[\sum_k^p\left(H_k^i\rho_k^i T_k^i\Delta\Phi_k^i + T_c\Delta T\right)\right] - Q_e = 0 \tag{2.6}$$

where:

- $V(x, y, z)$ is the volume of the block
- $T_k^i = T^i \frac{k_{rk}^i}{\mu_k^i}$ is the phase transmissibility
- $T^i(x, y, z)$ - the constant geometric part of the transmissibility
- $\Delta\Phi_k^i = p_k^a - p_k^b - (\rho_k^a - \rho_k^b)gz$ is the difference in phase potential of block a and b surrounding the interface i
- T_c^i is the thermal conduction transmissiblity
- ΔT^i is the difference in temperature between blocks a and b between the interface i

The two-point flux approximation (TPFA) method was used to discretize the spatial dimension for all the numerical solutions in this study. It is important to note that AD-GPRS is capable of multipoint flux approximation. This is important in geothermal applications when accurately representing the complex geology, which is pertinent in the shapes of fracture dominated flows. The TPFA implemented in this study is a phase-based, single-point upstream weighted scheme (Aziz and Settari, 1979).

The Newton-Raphson method was used to linearize the system of nonlinear equations into a system of linear equations. This requires a Jacobian matrix to be computed and solved for during each iteration. The Jacobian matrix is the derivatives of each nonlinear equation with respect to each individual variable. Other than the conservation equations (Equation 2.1 and 2.2) all other relationships are constraints that only pertain to a local block. For the mass and energy conservation equations, the above TPFA scheme is used to relate the properties between neighboring blocks.

To reduce the size of this entire global linear system a Schur-complement method was applied to the entire full Jacobian of each block such that only the mass and energy conservation equations are expressed as a function of the primary variables. The general Schur-complement strategy is described in Cao (2002). To illustrate this

briefly let the following equations be the full system of equations for a single block:

$$\begin{bmatrix} J_{pp} & J_{ps} \\ J_{sp} & D \end{bmatrix} \begin{bmatrix} J_{ss} \\ X_s \end{bmatrix} = \begin{bmatrix} R_p \\ R_s \end{bmatrix} \tag{2.7}$$

where:

- J_{pp} are the derivatives of the primary equations with respect to the primary variables
- J_{ps} are the derivatives of the primary equations with respect to the secondary variables
- J_{sp} are the derivatives of the secondary equations with respect to the primary variables
- J_{ss} are the derivatives of the secondary equations with respect to the secondary variables
- X_p are the primary variables
- X_s are the secondary variables
- R_p are the primary residual equations
- R_s are the secondary residual equations

When applying the Schur-complement, the system of equations for each block will now become:

$$\begin{bmatrix} J_{pp} - J_{ps}(J_{ss}^{-1}J_{sp}) & 0 \\ J_{ss}^{-1}J_{sp} & I \end{bmatrix} \begin{bmatrix} X_p \\ X_s \end{bmatrix} = \begin{bmatrix} F_p - (J_{ps}J_{ss}^{-1})F_s - J_{ps}J_{ss}^{-1}F_s \\ F_s \end{bmatrix} \tag{2.8}$$

With this, we can solve immediately for X_p and then back-calculate for X_s.

The Schur-complement allows for a flexible treatment of these primary variables and does not require any special choice of primary or secondary variable except that the Schur-complement matrix (J_{ss}) is nonsingular. This reduction is done on the full Jacobian of each block in the reservoir. With this reduced Jacobian, the reduced linear system of equations are solved and the primary variables are solved using a specified linear solver.

Once the primary variables have been solved, all other secondary variables can be back-calculated using the secondary equations that are locally defined on each block. Checks were implemented on this solution to ensure that each parameter remains within the physical range. This involves testing that the pressures, temperatures and enthalpy are positive and still within the applicable thermodynamic range of the thermodynamic relationships. It is also possible to implement various local chopping methods of these updates to guide the nonlinear solver to not take too large nonlinear steps. A specific local chopping method relating to phase transitions is discussed in Section 2.5.3.

An integral aspect of a nonlinear solver is the time-step selection scheme. This involves a balance between choosing a time step that is small enough such that the nonlinear solver would still converge and large enough such that the simulation could be computed in a reasonable time frame. If the nonlinear solver does not converge, typically what is done is the time step is repeated but with a smaller time step size . The strategy implemented in this study follows this idea. More formally the strategy is as follows, if the nonlinear solver converges after a specified time step, the following time step size is increased by a fixed multiplier. If the nonlinear solver does not converge, the time step size is reduced by dividing it by a fixed constant. If the time step reaches a minimum size, the simulation would stop. Convergence of a nonlinear step is defined based on the norm of the residual:

$$\max_{i,j} |r_{i,j}| < \epsilon \tag{2.9}$$

where r_j is the normalized residual of the j-th equation for the i-th block and ϵ is a specified tolerance value (in most large simulations $\epsilon = 10^{-4}$).The residuals for the

mass and energy conservation equation are normalized by the total mass and the total energy in the block.

2.3 Primary Variables

A fundamental step in the formulation of the numerical solution for these conservation equations is the choice of primary variables that are solved at each linear step. The choice of the primary variables will alter the Jacobian and the derivatives that are calculated with respect to each variable, thus altering the Newton convergence paths.

Although the choice of primary variables is nonunique, the primary variables have to fully define the thermodynamic state of the system. From Gibb's phase rule we know that the number of degrees of freedom to fully define the intensive thermodynamic variables is:

$$f = N_c + 2 - N_p \tag{2.10}$$

where:

- N_c is the number of components
- N_p is the total number of phases
- f is the number of degrees of freedom to define the intensive thermodynamic variables

There are also $N_p - 1$ saturation degrees of freedom. Thus the total number of degrees of freedom (N_f) is:

$$N_f = (N_c + 2 - N_p) + (N_p - 1) = N_c + 1 \tag{2.11}$$

For the geothermal problem of single-component water and a two-phase system, there are two primary variables to be solved for each block of the system. This is equal to the number of mass balance equations (N_c) and the energy balance equations (1) per block. There are a number of different combinations that could be chosen as the

primary variables for these simulations. This study compared these two main primary variable choices, a pressure-enthalpy formulation and a pressure-temperature-saturation formulation. The main difference between these two primary variable choices is that for the pressure-enthalpy formulation, pressure and enthalpy completely define the thermodynamic state for both single- and two-phase conditions. However for the pressure, temperature and saturation formulation, a variable switching process is required when switching between single and two-phase conditions, because pressure and temperature are no longer independent at saturated conditions.

2.4 Pressure-Temperature-Saturation Formulation

Table 2.1: pTS formulation status table for liquid water and steam

	Phases		Primary Variables		
Status	Liquid Water	Steam	p	T	S_s
0	x	-	x	x	-
1	-	x	x	x	-
2	x	x	x	(x)	x

The pressure, temperature and saturation formulation (pTS) follows the natural variable formulation developed by Coats (1980b). This formulation will be referred to as the pTS formulation. For the pTS formulation, pressure, temperature and the steam saturation are chosen as primary variables. However, the choice depends on the status of a block. A status is defined to be the composition and phase state of the block. For a pure water two-phase system, there are only three statuses, pure liquid water, pure superheated steam and a two-phase system. In single-phase conditions (compressed water or superheated steam) pressure (p) and temperature (T) are

the primary variables. However in two-phase conditions, pressure and temperature are not independent but now related by $p = p_{sat}(T)$, thus p and S_s are the primary variables in the two-phase region. A summary of this is shown in Table 2.1.

2.4.1 Phase Change Algorithm

Based on the current status of a block, the logic for determining the new status of a block after each Newton update is shown in Algorithm 1. It can be seen that in each phase transition, the temperature is preserved from the update rather than the pressure from the last iteration. This was selected because in most flow problems, the temperature tends to be more slowly varying than pressure. Here ϵ refers to a small number of order 10^{-4}, changing this value could affect the results of the solution as observed in Hernandez et al. (2015).

Algorithm 1 Phase behavior logic for pTS formulation

```
if status = 0 (Liquid Water) then
    if p < p_sat then
        status = 2
        p = p_sat(T)
        S_s = ϵ
    end if
else if status = 1 (Steam) then
    if p > p_sat then
        status = 2
        p = p_sat(T)
        S_s = 1 − ϵ
    end if
else if status = 2 (Two-phase) then
    if S_s > 1 then
        status = 1
    else if S_s < 0 then
        status = 0
    end if
end if
```

2.5 Pressure-Enthalpy Formulation

The pressure-enthalpy formulation (ph) presented here follows the study done by Faust and Mercer (1979). The formulation involves using pressure and enthalpy as the primary variables for all statuses. For this formulation, the primary variables remains unchanged throughout all the different statuses as shown in Table 2.2.The status of a block refers to the same definition as defined in the pTS formulation in 2.4. This formulation will be referred to in later sections as the ph formulation for brevity.

Table 2.2: ph formulation status table for liquid water and steam

	Phases		Primary Variables	
Status	Liquid Water	Steam	p	h
0	x	-	x	x
1	-	x	x	x
2	x	x	x	x

2.5.1 Phase Behavior

For single-phase conditions, the enthalpy (h) represents the single-phase fluid enthalpy. For the two-phase region, h represents the enthalpy of the fluid mixture defined as:

$$h = \frac{S_w \rho_w h_w + S_s \rho_s h_s}{\rho} \tag{2.12}$$

where:

- $S_{w/s}$ is the saturation of water/steam
- $\rho_{w/s}$ is the saturated density of water/steam
- $h_{w/s}$ is the saturated enthalpy of water/steam

and the density of the mixture ρ is defined to be:

$$\rho = S_w \rho_w + S_s \rho_s \tag{2.13}$$

from the saturation constraint the saturation is related by:

$$S_w + S_s = 1 \tag{2.14}$$

From Equations 2.12, 2.13 and 2.14, we can rearrange to obtain an equation for S_w and thus S_s,

$$S_w = \frac{\rho_s(h_s - h)}{h(\rho_w - \rho_s) - (h_w\rho_w - h_s\rho_s)} \tag{2.15}$$

2.5.2 Basic Phase Change Algorithm

To account for the phase change between statuses, the enthalpy value was compared with the saturated enthalpies for water and steam respectively. The phase change logic is based upon Equation 2.15, where if $h > h_s \Rightarrow S_w < 0$ resulting in superheated steam and if $h < h_w \Rightarrow S_w > 1$ meaning a liquid water state. Although there is no change in the primary variables for these different statuses, the phase transition still needs to occur as the thermodynamic relationships will depend on the status of the block. This difference is mainly the definition of what is the phase enthalpy and phase density, because for the single-phase conditions, the phase enthalpy would be the enthalpy of the entire fluid but for two-phase conditions, the phase enthalpy is the saturated enthalpy of the fluid. The algorithm for the phase change logic can be seen in Algorithm 2. When transitioning into a new phase, the water saturation can be calculated as a function of pressure and enthalpy based on (Equation 2.15).

For this phase-change algorithm, the pressure and enthalpy were updated based on the nonlinear update from Newton's method. It was found that if the phase change was conducted in this way, the nonlinear solution would find difficulty in converging when a phase transition occurs. This was particularly so for the cases where the converged solution was close to the phase boundaries ($S_w = 1 - \epsilon$ or $S_w = \epsilon$ where ϵ is a small number ($\approx 10^{-2}$)). For these cases if this phase change algorithm was implemented, there would be oscillation between the two different statuses due to the nonlinearity of the solution. To overcome this, a chopping type method was implemented when transitioning to the two-phase region as described in the following section.

Algorithm 2 Basic Phase behavior logic for ph formulation

if status = 0 (Liquid Water) **then**
  if $h > h_w(p)$ **then**
    status = 2
    $S_w(p,h) = ...$ (Equation 2.15)
    $S_s = 1 - S_w$
  end if
else if status = 1 (Steam) **then**
  if $h < h_s(p)$ **then**
    status = 2
    $S_w(p,h) = ...$ (Equation 2.15)
    $S_s = 1 - S_w$
  end if
else if status = 2 (Two-phase) **then**
  if $h > h_s(p)$ **then**
    status = 1
  else if $h < h_w$ **then**
    status = 0
  end if
end if

2.5.3 Phase Behavior Update with Chop Algorithm

A chopping algorithm for the enthalpy was implemented when crossing between the phase boundaries. This algorithm chopped the nonlinear update close to the phase boundary rather than letting the updated pressure and enthalpy to define the new saturation of the mixture. This chopping algorithm was based upon the phase update algorithm implemented by Pruess et al. (1999) for the pTS formulation. First the temperature is calculated as a function of the new pressure and enthalpy updates, from this temperature the saturation pressure is calculated to be the pressure at the next nonlinear step. Based on this saturation pressure, the saturated enthalpy is

calculated and the enthalpy for the next nonlinear step is set to be slightly greater than the saturated water enthalpy or slightly less than the saturated steam enthalpy. This slight increase or decrease is specified based on a specified tolerance denoted here as ϵ, where ϵ could be adjusted and could affect the nonlinear convergence for different scenarios. By default the tolerance is set to be $\epsilon = 10^{-5}$. This ϵ could also be set such that it would be consistent with the pTS formulation so that for both cases, each time a phase change occurs the same update would occur. This also allows the phase change logic to be the same as the approach to the pTS formulation as shown in Algorithm 1. This would allow for the comparison between both formulations to be independent of the phase transition logic and purely dependent on the derivatives and nature of each formulation.

Algorithm 3 Phase behavior logic for ph formulation with chopping at the phase boundaries

if status = 0 (Liquid Water) **then**
  if $h > h_w$ **then**
    status = 2
    $T = T(p, h)$
    $p = P_{sat}(T)$
    $h = h_w(p)(1 + \epsilon)$
    $S_w = ...$ (Equation 2.15)
    $S_s = 1 - S_w$
  end if
else if status = 1 (Steam) **then**
  if $h < h_s$ **then**
    status = 2
    $T = T(p, h)$
    $p = P_{sat}(T)$
    $h = h_s(p)(1 - \epsilon)$
    $S_w = ...$ (Equation 2.15)
    $S_s = 1 - S_w$
  end if
else if status = 2 (Two-Phase) **then**
  if $h > h_s$ **then**
    status = 1
  else if $h < h_w$ **then**
    status = 0
  end if
end if

2.6 Thermodynamic Relations

To conduct a comparison between the two formulations, a consistent set of thermodynamic relations must be formulated for both cases. For simplicity, polynomial fits were implemented, these were identical to those used in Faust and Mercer (1979). These thermodynamic parameters were developed in that study for the ph formulation, and were slightly adapted for the pTS formulation such that consistency with the ph formulation could be achieved.

2.6.1 Thermodynamic Relations for ph Formulation

Faust and Mercer (1979) performed least square regression on experimental data describing the thermodynamic parameters of water. In this study, these identical relationships were implemented for the enthalpy formulation. The phase viscosities were assumed to be functions of temperature:

$$\mu_s = 10^{-4}(0.407(T - 273.15) + 80.4) \tag{2.16}$$

$$\mu_w = 10^{-4}\left(241.4 \times 10^{247.8/((T-273.15)+133.15)}\right) \tag{2.17}$$

where:

- μ_s [cp] is the dynamic viscosity of steam
- μ_w [cp] is the dynamic viscosity of water
- T is the temperature of the fluid in [K]

The saturated steam enthalpy, h_s and saturated water enthalpy, h_w, were defined as functions of pressure:

$$\begin{aligned} h_s =& 10^{-7}[2.82282 \times 10^{10} - 3.91952 \times 10^{5} p^{-1} \\ &+ 2.54342 \times 10^{21} p^{-2} - 9.38879 \times 10^{-8} p^{2}] \end{aligned} \tag{2.18}$$

$$\begin{aligned} h_w =& 10^{-7}[7.30984 \times 10^9 + 1.29239 \times 10^2 p - 1.00333 \times 10^{-6} p^2 \\ &+ 3.9881 \times 10^{-15} p^3 - 9.90697 \times 10^1 5 p^{-1} + 1.29267 \times 10^{22} p^{-2} \\ &- 6.28359 \times 10^{27} p^{-3}] \end{aligned} \tag{2.19}$$

where:

- p is in [dynes/cm^2] or 10^{-6} [bar]
- h_s and h_w are in [kJ/kg].

For the ph formulation, temperature was treated as a function of pressure and enthalpy in the single-phase regions. For compressed water:

$$T = 273.15 - 2.41231 + 2.56222 \times 10^{-8} h - 9.31415 \times 10^{-17} p^2 - 2.2568 \times 10^{-19} h^2 \tag{2.20}$$

and for the superheated-steam region:

$$\begin{aligned} T =& 273.15 - 374.669 + 4.79921 \times 10^{-6} p - 6.33606 \times 10^{-15} p^2 + 7.39386 \times 10^{-19} h^2 \\ &- 3.3372 \times 10^{34} h^{-2} p^{-2} + 3.57154 \times 10^{19} p^{-3} - 1.1725 \times 10^{-37} h^3 p - 2.26861 \times 10^{43} h^{-4} \end{aligned} \tag{2.21}$$

where:

- T is in [K]
- p is in [dynes/cm^2] or 10^{-6} [bar]
- h is in [ergs/g] or 10^-7 [kJ/kg]

For the two-phase region, the saturated temperature, h_w was used instead of h in Equation 2.20. In the single-phase regions, the steam and water densities were chosen

to be functions of pressure and enthalpy. For the compressed-water region:

$$\begin{aligned}\rho = \rho_w =& 10^3[1.00207 + 4.42607 \times 10^{-11}p - 5.47456 \times 10^{-12}h \\ & 5.02875 \times 10^{-21}hp - 1.24791 \times 10^{-21}h^2]\end{aligned} \tag{2.22}$$

and for the superheated steam region:

$$\begin{aligned}\rho = \rho_s =& 10^3[-2.26162 \times 10^{-5} + 4.38441 \times 10^{-9}p - 1.79088 \times 10^{-19}ph \\ & + 3.69276 \times 10^{-36}p^4 + 5.17644 \times 10^{-41}ph^3]\end{aligned} \tag{2.23}$$

where ρ is in [kg/m^3]. For the two-phase regions where the saturated water and steam densities are required, the saturation pressures and enthalpies are used in Equations 2.22 and 2.23 to compute ρ_w and ρ_w.

2.6.2 Thermodynamic Relations for pTS Formulation

To be consistent with the thermodynamic relationships used in the p-h formulation, h must be computed in a manner that is consistent with the above formulations. Once h is calculated, all other thermodynamic properties could be computed as shown in Section 2.6.1. In the single-phase regions, Equation 2.20 and 2.21 could be used to compute h. Equation 2.20 is a quadratic equation with h, thus knowing p and T, h could be calculated directly:

$$h(p,T) = \frac{-B + \sqrt{B^2 - 4D\left(A + Cp^2 - (T + 273.15)\right)}}{2D} \tag{2.24}$$

where the constants $A = -2.41231$, $B = 2.5622 \times 10^{-8}$, $C = -9.31415 \times 10^{-17}$ and $D = -2.2568 \times 10^{-19}$, where p is measured in [dynes/cm^2] or 10^{-6} [bar] and T is in [K]. However, for the superheated steam case (Equation 2.21), an explicit solution is not available for h. A bisection method using Equation 2.21 was used to compute h from p and T. The starting guesses for the bisection was chosen to be h_s (at the pressure calculated) and 3200 [kJ/kg] (the maximum range of the thermodynamic parameters). This bisection algorithm is shown in detail in Algorithm 4.

For the two-phase region, the saturated fluid properties (Equation 2.18, 2.19, 2.22, 2.23) are calculated directly from from p. Here, h is calculated directly using these saturated properties and S_s (Equation 2.12).

Algorithm 4 Bisection method for computing the superheated steam as a function of pressure and temperature for pTS formulation

function computeSuperheatedSteamEnthalpy(p, T)
$a = h_s(p)$ ▷ Initial lower bound is the saturated steam enthalpy
$b = 3200\text{kJ/kg}$ ▷ Initial upper bound is max value of the thermodynamic formula
$N = 1$
while $N \leq N_{max}$ **do**
  $c = \frac{a+b}{2}$ ▷ New Midpoint
  tempC $= T(p, c)$ ▷ $T(p, h)$ = Equation2.21
  if $|T - \text{tempC}| < \epsilon$ or $\frac{b-a}{2} < \epsilon$ **then**
    break
  end if
  $N = N + 1$
  if tempC $< T$ **then** ▷ Update Bounds
    a = c
  else
    b = c
  end if
end while
if $N > N_{max}$ **then**
  Output("Bisection Failed")
end if
return c
End Function

2.7 Concluding Remarks

The full mathematical formulation pertaining to geothermal reservoirs was presented in this chapter. An introduction to the typical reservoir simulation solution process was presented. In particular, the choice of primary variables and how it is implemented in the solution process of a simulator was presented. Based on this solution process, two variable formulation approaches were presented; a pTS formulation with pressure, temperature and saturation as the primary variables and an enthalpy formulation with pressure and enthalpy as the primary variables. In order for further comparisons between these two formulations, a consistent implementation of the thermodynamic and phase behavior was discussed and implemented. These two formulations would serve as the basis of the implementation using the AD-GPRS framework.

Chapter 3

AD-GPRS

This chapter describes the AD-GPRS (Automatic Differentiation based General Purpose Research Simulator) framework developed at the SUPRI-B Research Group at Stanford University and how it was modified for the geothermal formulations described in Chapter 2. Due to the complexity of AD-GPRS, only the aspects of the framework that pertain to the implementation of the geothermal fluid module will be described. A full comprehensive description of the code is available in Voskov et al. (2012).

AD-GPRS was designed to be a flexible reservoir research simulator that focuses on an extendable code structure for investigating different physics and numerical problems. Consequently, a modular object-orientated code structure was implemented. This flexible framework was crucial when implementing the geothermal module to reduce the amount of code duplication. AD-GPRS solves the governing mass and energy balance equations, by discretizing with the finite volume method and linearizing the nonlinear equations using a Newton-Raphson based method. The entire code was written in standard C++. The nonlinear framework of AD-GPRS was built on the Automatic Differentiation Expression Template Library developed at Stanford University (Younis, 2011; Zhou, 2012).

3.1 Numerical Framework Based on ADETL

This flexible structure of the code architecture is driven by the use of Automatic Differentiable Expression Templates Library (ADETL). ADETL is a large optimized C++ library that consists of C++ data structures and optimized algorithms that perform automatic differentiation. ADETL provides the foundation of the code architecture for AD-GPRS and determines how the data structures and variables are handled. The main benefit of using this library is that it allows for the automatic generation of sparse analytical Jacobians from the definition of any nonlinear residual equation. This process requires no guidance from the user and only requires the formulation with the specified unknowns. Thus, for the implementation of the geothermal module, only changes to the computation of the residual have to be modified and the Jacobian of the new residual is calculated automatically. This flexibility is particularly important when solving with different primary variable approaches and implementing different thermodynamic relationships for water. This is because for different variable formulations, a manual implementation would require the Jacobian to be rewritten. This difficulty is intensified if different thermodynamic relationships are experimented with, because these relationships can be complex and obtaining manual derivatives is difficult.

3.1.1 ADETL Implementation

Code 3.1: Example of using ADETL structures

```
1 // Declaring the independent variables
2 ADscalar<> a = 4;
3 ADscalar<> b = 3
4 // Write Nonlinear Residual equation
5 ADscalar<> c = 0.5*a + b*b;
6 // Output will calculate derivatives
7 cout << c << endl; // output: 11 [0.5 6];
```

The foundation of ADETL's implementation is through the use of C++ expression templates, data structures and overloading of all the basic operators. These data structures store both the function values and the derivative with respect to the independent variables. Using these overloaded operators, when evaluating a certain expression, both the values and derivatives are computed. Finally, the extracted Jacobian from the nonlinear residual, is stored in a Compresssed Sparse Row (CSR) format.

The basic underlying method to compute these derivatives involves:

1. Analyzing the expressions to determine the basic operators (+,-,*,/) that are involved

2. Performing basic differentiation rules – linearity, product, quotients, function derivatives and chain rule to evaluate the derivative

An example of how this library was implemented is shown as Code 3.1. For this example, the derivatives of a basic equation defined as:

$$c = 0.5a + b^2$$

where a and b are the independent variables initialized as: $a = 4$ and $a = 3$. The derivatives and the value of this value are calculated automatically as shown in the example.

3.1.2 Inverse Method

Due to the complexity and the nature of the thermodynamics in multiphase fluids, often an explicit relationship for all the fluid properties cannot be obtained. For a single-component case, basic polynomial curve fits are able to represent fluid properties explicitly. However for a multicomponent case, (if CO_2 was added) an Equation of State or iterative approach might be needed. These iterative methods mean that the ADETL data structures would not contain the correct derivatives. To overcome this problem, the inverse function theorem was applied such that the derivatives are calculated implicitly through other fluid relationships.

For the geothermal pTS formulation (Section 2.6.2), even for simplified polynomial thermodynamic relationships, an iterative bisection approach (Algorithm 4) was required to compute the fluid enthalpy from the pressure and temperature value. This was to maintain consistency between the thermodynamic relationships between the pTS and ph formulations. To evaluate the associated derivatives, the Inverse Theorem Method was used for these cases. As an example, when computing the superheated steam fluid enthalpy (h) with respect to the pressure (p) and temperature (T), the equation is as follows:

$$\begin{aligned} T(p,h) =& 273.15 - 374.669 + 4.79921 \times 10^{-6} p - 6.33606 \times 10^{-15} p^2 + 7.39386 \times 10^{-19} h^2 \\ & - 3.3372 \times 10^{34} h^{-2} p^{-2} + 3.57154 \times 10^{19} p^{-3} \\ & - 1.1725 \times 10^{-37} h^3 p - 2.26861 \times 10^{43} h^{-4} \end{aligned} \tag{3.1}$$

However, there is no simple closed-form solution to obtain $h(p,T)$ based on this formulation. Instead, a bisection method was implemented as discussed earlier. Because a bisection method was used, the derivatives of $\frac{\partial T}{\partial p}$ and $\frac{\partial T}{\partial h}$ would be calculated using the inverse function method.

In general the inverse function method is as follows:
We first define a function **F(Z,X)**. For this example:

$$\mathbf{F}(\mathbf{Z}, \mathbf{X}) = h - h(p,T) = 0 \tag{3.2}$$

Here, $h(p,T)$, is the function (Equation 3.1) to compute h as a function of p and T, and h is the enthalpy of the fluid. The derivative can then be calculated using the inverse function method.

$$\frac{d\mathbf{Z}}{d\mathbf{X}} = \frac{d\mathbf{F}}{d\mathbf{X}} \left(\frac{d\mathbf{Z}}{d\mathbf{F}} \right)^{-1} \tag{3.3}$$

We can see that in general, for this method to be valid, we require $\mathbf{F}$ to be the same size as $\mathbf{Z}$ and $\mathbf{X}$ would be equal to be the number of primary variables. The inverse method is useful when investigating different thermodynamics and different variable formulations. This is especially true if an equation of state type formulation is used

to calculate the thermodynamic properties. The need to use these inverse method depends wholly on the thermodynamic relationships that are implemented and what their independent variables are.

3.1.3 ph Formulation Inverse Method

For the ph formulation, the inverse method was used to compute the derivatives of the saturation of water and the temperature with respect to the pressure and enthalpy. Although there are explicit formulas for temperature and saturation as a function of pressure and temperature, the inverse method was implemented in the event more complex and accurate thermodynamic relationships would be needed, where an explicit relationship would not be available. This would be true if we add an additional component and an equation of state or an iterative type method would be used.

For this formulation we have the following definitions:

$$\mathbf{Z} = \begin{bmatrix} S_w \\ T \end{bmatrix}, \mathbf{X} = \begin{bmatrix} p \\ h \end{bmatrix}, \mathbf{F} = \begin{bmatrix} h(\rho_w S_w + (1 - S_w)\rho_g) - (\rho_w S_w h_w + \rho_g(1 - S_w)h_s) \\ T - T(p, h) \end{bmatrix} \tag{3.4}$$

Here ρ_w (Equation 2.22) and ρ_g (Equation 2.23) are the saturated phase densities as functions of pressure and enthalpy, h_w (Equation 2.19) and h_g (Equation 2.18) are the saturated phase enthalpies, $T(p, h)$ is temperature as a function of pressure and enthalpy, S_w is the water saturation, T is the temperature, p is the pressure and h is the enthalpy of the fluid mixture. Thus by using these as our definitions for the Inverse Theorem Method, the derivatives of S_w and T can be obtained with p and h as the independent variables.

3.1.4 pTS Formulation Inverse Method

As discussed briefly in Section 3.1.2, when computing the single-phase fluid enthalpy (h) as a function of pressure (p) and temperature (T), an explicit formula was unavailable, thus the derivatives could not to be calculated directly. This is only the case for the single-phase situation, because in two-phase the enthalpy (h) of the fluid can be calculated explicitly from Equation 2.12. Thus, a check was included to test if the block was in single phase before the Inverse Method was calculated. In the single-phase regions, the definitions for the inverse method is as follows:

$$\mathbf{Z} = \begin{bmatrix} h \end{bmatrix}, \mathbf{X} = \begin{bmatrix} p \\ T \end{bmatrix}, \mathbf{F} = \begin{bmatrix} h - h(p,T) \end{bmatrix} \tag{3.5}$$

Here $h(p,T)$ is a function that calculates the fluid enthalpy (h) as a function of pressure (p) and temperature (T). This is the bisection method applied on Equation 2.21 and 2.24. From these definitions of the Inverse Theorem Method, the derivatives of h with respect to the primary variables p and T can be computed.

3.2 Programming Structure of AD-GPRS

The structure of AD-GPRS was designed to be flexible and to allow for additional physics and numerical schemes to be implemented and investigated easily. The code structure follows a modular object-orientated programming design. The entire detailed class structure of AD-GPRS and the code structure can be seen in Zhou (2012). This section will describe how the two geothermal formulations were implemented based on this modular structure. Only aspects pertaining to these changes are discussed. To be consistent with the convention used in Zhou (2012), *italic shape* will be used to denote class names, *slanted shape* to denote variable names, and **boldface** to denote function names.

One of the benefits of this modular structure is that it reduced the amount of code duplication in the implementation of the geothermal aspect into the current

AD-GPRS simulator framework. The two main changes that differentiate an oil and gas simulator and this geothermal formulation are the variable formulation and the thermodynamics involved with a pure water multiphase simulation.

3.2.1 NonlinearFormulation Object

The *NonlinearFormulation* abstract class currently has implemented a Natural Variable, Molar Variable and Gamma Variable formulation implementations and their thermal equivalents. These formulations are formulated for multicomponent and multiphase reservoir simulations. A key assumption for all of these existing formulations is that the number of phases is always greater than or equal to the number of components, which is the case for all oil and gas simulations.

However, for the geothermal simulations, there is only one component (water), which is less than the number of possible phases (liquid water and steam). Thus we were unable to use these preexisting formulations directly, thus two new formulations were written. These two new inherited classes of the *NonlinearFormulation* were constructed based upon the preexisting Natural Variable Formulation and Molar Variable Formulations. The pTS and ph formulations for the geothermal problem are analogous to the Natural Variable and Molar Variable Formulations, but were written specifically for the one-component two-phase geothermal system. We will refer to these new inherited classes as *NaturalVariableFormulationG* and *MolarVariableFormulationG*, where *G* represents the geothermal equivalent to the already existing classes.

One of the benefits of having two different formulations on a unified framework is that this allows for a fair comparison between the formulations. Often if a new formulation is investigated, it would require a large amount of code duplication and can be difficult to guarantee that difference in performance could be a result of other aspects of the simulator and not the formulation itself. Thus having the formulations separated in this manner will allow for a direct comparison of the numerical behavior, this behavior is discussed later in Chapter 5.

Implementation Details

NonlinearFormulation encompasses all the formulation specific aspects in the implementation of a reservoir simulator. These aspects are contained within the virtual member functions of the object that are called during both initialization and throughout the simulation.

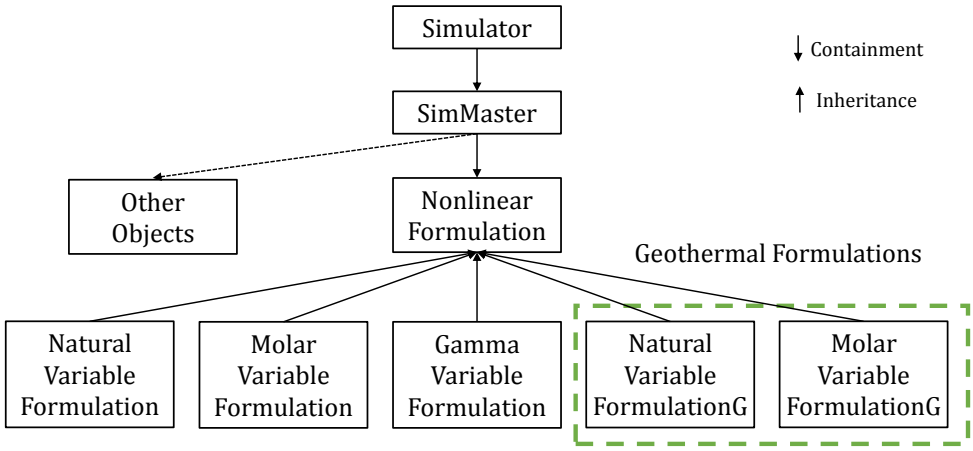

Figure 3.1: Structure of Nonlinear Formulation with respect to entire simulator including new inherited geothermal classes

In the initialization process the **makeStructure**, **buildStatusTable** and **convertInputToFormulation** functions are called. The procedure**makeStructure** creates a table of the different independent variable combinations associated with each different statuses, this specifies the independent variables for the different statuses as specified in Table 2.2 and 2.1. Another procedure **buildStatus** generates the table consisting of all the different phase combinations, this is identical for both the pTS formulation and ph formulation for the geothermal problem, because both formulations have the same number of possible statuses. Finally, **convertInputToFormulation**

takes the specified input variables from the user and converts them into the appropriate primary variables for that formulation. This allows the user to specify the same set of initial conditions for different formulations. If the user specifies a S_s such that $0 < S_s < 1$, it will be assumed that the block is at saturation conditions and the input of given a T will be ignored and will be set to saturated conditions ($T = T_{sat}(p)$).

The phase transition logic associated with each formulation is performed in the **changeStatus** function. This function would implement the phase change logic described in Algorithm 1 and 3. **changeStatus** takes the updated variables from the nonlinear solver and updates the status of the block. Once the status is determined, the fluid properties are calculated (**fluidPropertiesCaclualtion**) and all the saturation constraints are satisfied (**enforceConstraints**). The **fluidPropertiesCalculation** calls the fluid class that contains the thermodynamic formulations specific to the fluid. This follows the modular code structure by dividing the fluid property calculations and the formulation specific variables.

Another virtual function is **makeSecondaryTerm**, which constructs the secondary equations that are local to a certain block. For the enthalpy formulation, because there are only two variables and this does not change depending on the phase status, there are no any secondary equations. For the pTS formulation, the secondary equation appears when the block is at saturated conditions. The role of the secondary equation is to relate the secondary variables to the primary variables, for the pTS formulation case this is relating the temperature to the pressure at saturation conditions. This is analogous to the fugacity calculation in oil and gas simulations, where the fugacity constraints have to be satisfied. More formally, the secondary equation for the geothermal natural variable formulation is:

$$T - T_{sat}(p) = 0 \tag{3.6}$$

where:

- T is the temperature of the block
- T_{sat} is the saturation temperature as a function of the pressure p.

It is important to note that these functions are used in both the reservoir calculations and also for well fluid property calculations.

3.2.2 Fluid

For the geothermal implementation, the general *MultiPhaseCompFluid* was used to describe the geothermal fluid. Within the *Fluid* object, a vector of the *Phase* objects is contained. These contain the methods that calculate the thermodynamic formulations for each phase. This would allow the specification of different calculation methods for different phases. The full structure of this can be seen in Figure 3.2.

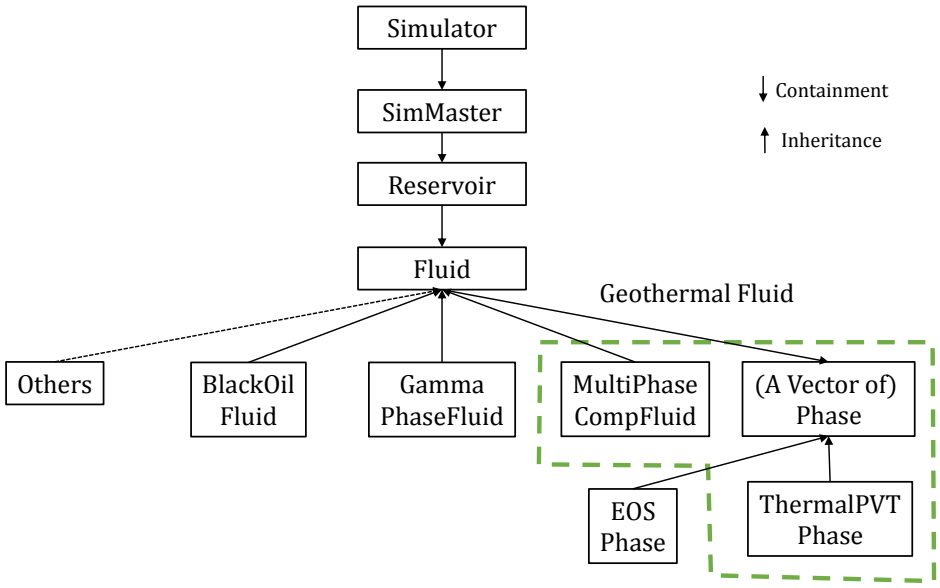

Figure 3.2: Structure of Fluid with respect to entier simulator including new inherited geothermal classes

As mentioned briefly earlier, all the thermodynamic relationships are able to be hidden from *Nonlinear Formulation* and are called from the *Fluid* class. All the density, enthalpy, viscosity and saturation pressure calculations are called from the *Fluid*

class. From the *Fluid* class, the specific formula is called from the *ThermalPVTPhase* class which is inherited from *Phase* which is a member of the *Fluid Class*.

3.2.3 Other Objects and Implementation details

One additional feature added to AD-GPRS was how wells were specified. The typical convention in oil and gas simulations is to specify well rates by volume flow rates, however the convention in geothermal simulations is for the wells to be specified by the mass flow rate. This slight difference allows a convenient definition for wells if geothermal reservoir models were implemented.

Other than *NonlinearFormulation* and the *ThermalPVTPhase*, no other major changes or additions to the code were required. This shows the modular and flexible nature of the AD-GPRS framework and how a different fluid structure and formulation was able to be incorporated without much code duplication. This is promising for the future, as other variable formulations such as internal density and enthalpy could also be implemented and tested.

3.3 Concluding Remarks

A geothermal reservoir simulator was implemented using the AD-GPRS simulator framework. The AD-GPRS framework has a flexible and modular code structure that utilizes the ADETL library. This flexible framework allowed the incorporation of a geothermal module that only needed the addition of two inherited objects from the *NonlinearFormulation* and *Fluid* classes. This sort of structure allows for a fair comparison between both formulations because all other aspects of the simulator would remain the same.

Chapter 4

Numerical Verification

To verify the simulation results obtained from AD-GPRS, a code comparison study was completed based on the results from the 1980 Code Comparison completed at Stanford (Stanford Geothermal Program, 1980). The results of both formulations was tested for consistency for these different test cases. The solutions were tested against analytical, semianalytical and TOUGH2 for the varying complexity of each test case. Without a verification of the results of each formulation, a comparison between the performance of each formulation would be difficult to quantify with confidence.

4.1 Single-phase Injection

4.1.1 Problem Specification

This test case (Test Case 1) involves a one-dimensional, radial steady-state flow with unsteady heat transport for single-phase liquid water. The objective of this test case is to verify the heat conduction and convection of single-phase liquid water.

This problem involves the injection of $433.15K$ liquid water at a flow rate of $q = 10$kg/sec into a reservoir of $444.15K$ water. The boundary condition imposed is a pressure of 50 bars at the outer radius of 1000m. The initial condition is that the entire reservoir is at a temperature of $441.5K$. The model properties are as follows:

Table 4.1: Model parameters for Test Case 1

Rock Permeability (md)	1000
Rock Density (kg/m^3)	2500
Rock Specific Heat Capacity (J/gK)	1
Reservoir Thickness (m)	100
Porosity	0.12
Spatial Grid	Uniform 25 m radial grid spacing

The results from this test case can be compared with the Avdonin analytical solution (Avdonin, 1964). A match for this case would verify that simulator is solving the mass and energy balance equations sufficiently and also handling the units and treatment of a constant mass injection rate correctly.

4.1.2 Results

Figure 4.1 shows the spatial temperature distribution at the final time of 10^9 seconds and the temporal temperature distribution at each time step for the block with its center at 37.5 m. Figure 4.2 shows the same results as Figure 4.1 but with the published results obtained in the 1980 Stanford Code Comparison study. As it can be seen from the results, AD-GPRS compares well with the analytical solution. The temperature distribution over time for the block at 37.5 m is slightly greater than the analytical Advonin solution. This deviation can be seen similarly in the Code Comparison Study except for the single code that managed to match the results. It is important to note that in that simulator, they implemented a central-differencing scheme rather than the traditional upwinding scheme. This suggests that the deviation between the two results could be from the error of approximation, since a central-difference has a second order of approximation, while the upwinding scheme is first order. However an upwinding scheme is used to guarantee that the entropy

solution is reached for hyperbolic transport.

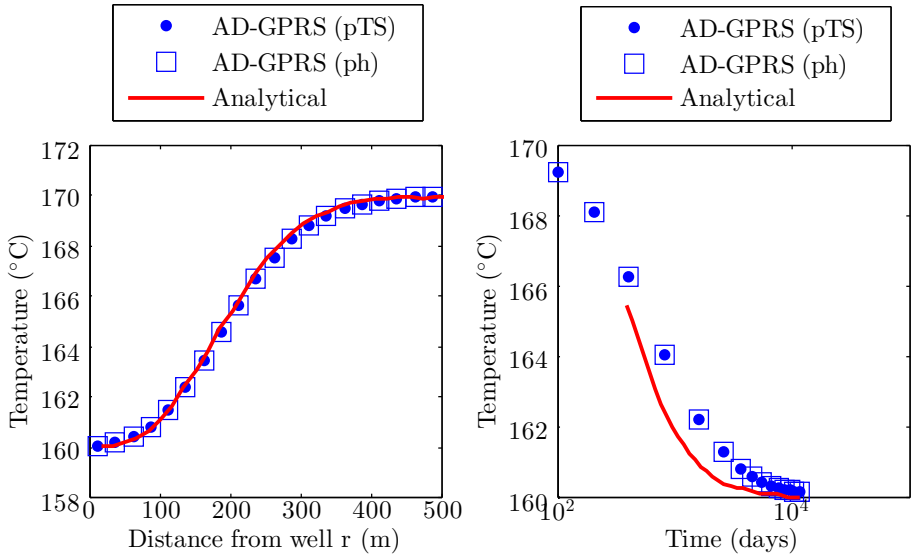

Figure 4.1: Temperature distribution at time = 10^9 seconds (left) and the temperature distribution corresponding to the block with a block center of 37.5m (right) over the simulated time period

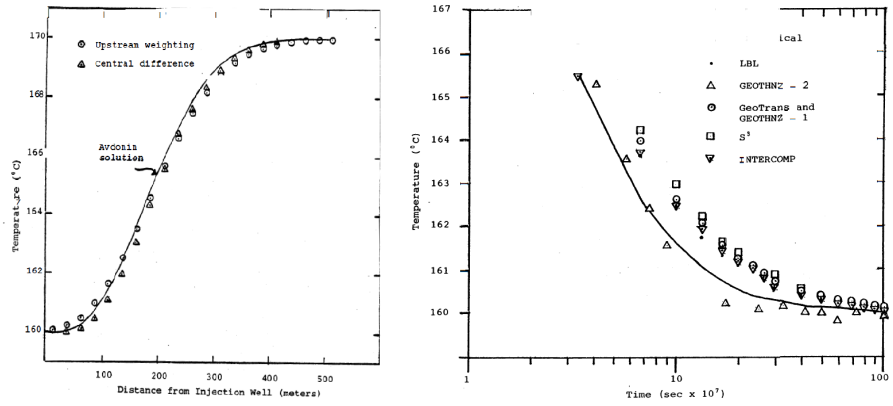

Figure 4.2: Results from the 1980 Stanford Code Comparison Study Temperature distribution at time = 10^9 seconds (left) and the temperature distribution corresponding to the block with a block center of 37.5m (right) over the simulated time period

4.2 Production

The following test cases involves three different production scenarios. Each of these test cases involves a constant mass production of fluid in a one-dimensional radial reservoir. The difference between each test case is that each production scheme results in a different fluid mixture and flow regime. The test cases are as follows:

(A) The reservoir remains liquid water throughout the production scheme

(B) The reservoir begins at a two-phase mixture of liquid water and steam and remains a two-phase mixture through the entire simulation period

(C) The reservoir begins as liquid water and changes to a two-phase mixture, and a flash-front propagates throughout the reservoir

Table 4.2: Model parameters for Test Case 2

	Case A	**Case B**	**Case C**
Initial pressure (bars)	90	30	90
Initial temperature (K)	533.15	506.95	573.15
Initial water saturation	1.0	0.65	1.0
Porosity	0.2	0.15	0.2
Permeability (md)	10	240	10
Thickness (m)	100	100	100
Producing Rate (kg/s)	14.0	16.7	14.0
Rock Compressibility	0	0	0
Rock thermal conductivity	0	0	0

The Corey Curves were used for the relative permeabilities relationships:

$$k_{rw} = (S^*)^4 \tag{4.1}$$

$$k_{rs} = \left((1 - S^*)^2 \left(1 - (S^*)^2\right)\right) \tag{4.2}$$

where:

$$S^* = \frac{S_w - 0.3}{0.65} \tag{4.3}$$

where:

- k_{rw} is the relative permeability of the water
- k_{rs} is the relative permeability of the steam
- S_w is the saturation of the fluid

The same spatial discretization was used as in the 1980 Code Comparison Study. The radius of each radial segment was defined to be:

$$r_n = 0.52^{\frac{n-1}{2}}[\text{m}], n = 1, 2, 3, ..., 26 \tag{4.4}$$

where r_n is radius of the block center for the block numbered n, the index 1 refers to the block closest to the well and 26 at the outer most radius block.

4.2.1 Results

Each of the different test cases for these production scenarios have either an analytical or semianalytical solution to the problem. The semianalytical solutions were solved using a similarity variable of t/r^2 (O'Sullivan, 1981) and transforming the partial differential equations to a system of ordinary differential equations that were solved numerically. To verify the results obtained from AD-GPRS, these analytical and semianalytical solutions were compared as a function of t/r^2.

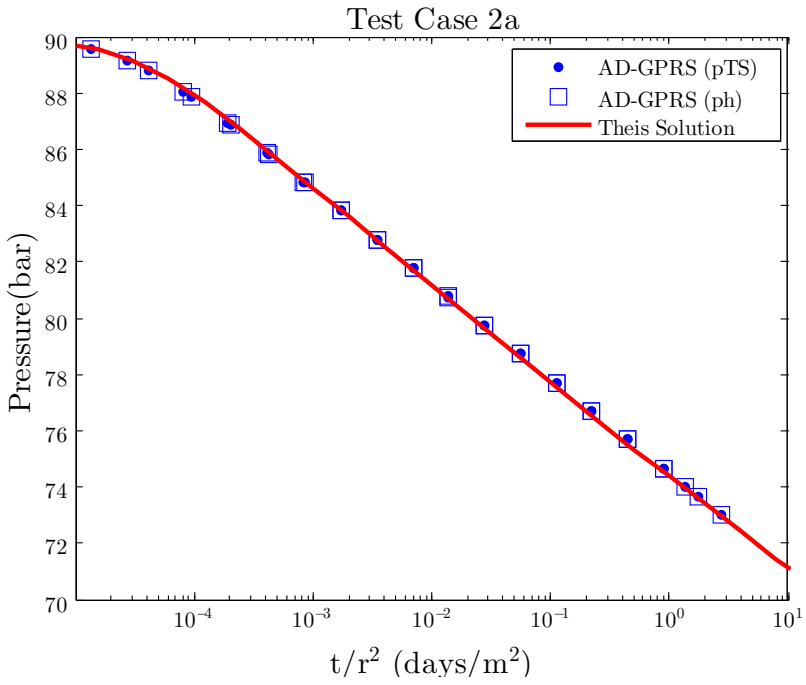

Figure 4.3: Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m) (Case 2A)

2a) Single-phase

For the single-phase case, the Theis solution (Theis, 1935) was used for comparison. Figure 4.3 shows the pressure as a function of t/r^2. As can be seen, there is a very good match between AD-GPRS and the Theis solution. Figure 4.3 was plotted using the results for the blocks at a block center of 0.5 m and 0.707 m, the pressure results at each time step for these blocks were plotted against the analytical solution. This agreement shows that the simulator is able to model the production from a well and the pressure changes associated with it.

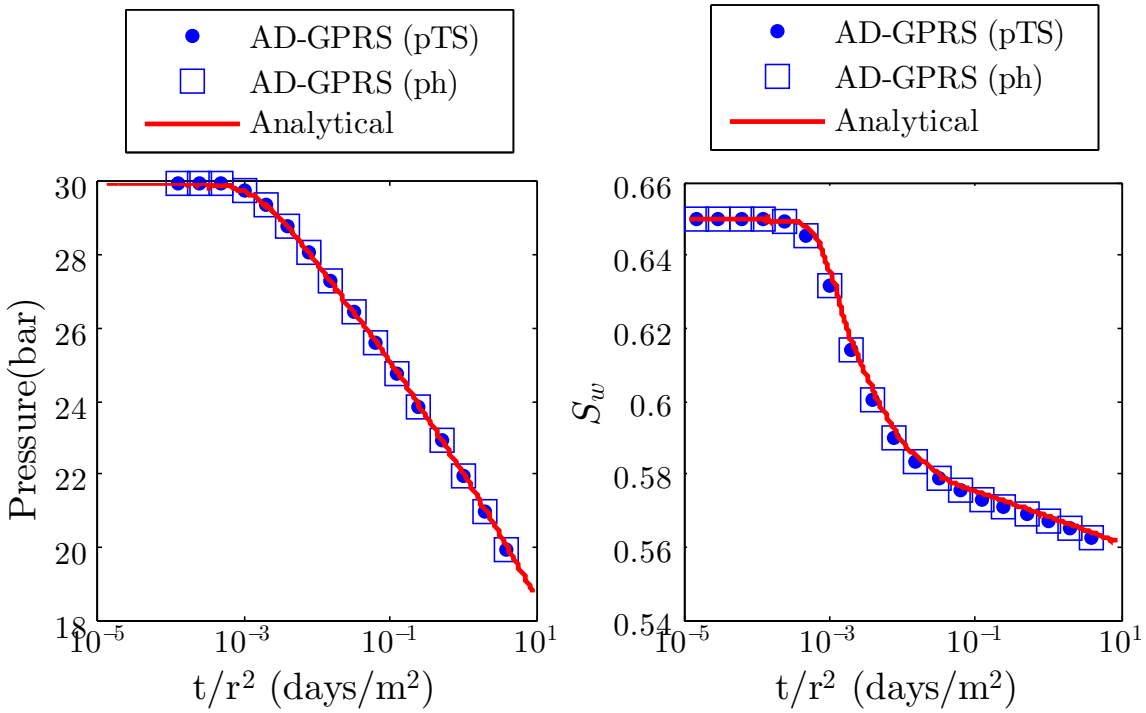

Figure 4.4: Pressure (left) and water saturation (right) plotted with t/r^2 for the pure two-phase case (Case 2B)

Two-phase

For this case, an analytical solution was not available for comparison. However instead, a semianalytical solution was used from O'Sullivan (1981). This similarity solution transforms the governing mass and energy balance partial differential equations into ordinary differential equations and integrates the ordinary differential equations numerically. Figure 4.4 shows an acceptable match between the semianalytical results and AD-GPRS. This slight deviation between the two results can be attributed to numerical dispersion. This can be seen by the smearing of the shock front. The results from the 1980 Code Comparison Study (Figure 4.5) shows a similar mismatch between the numerical and the semianalytical results.

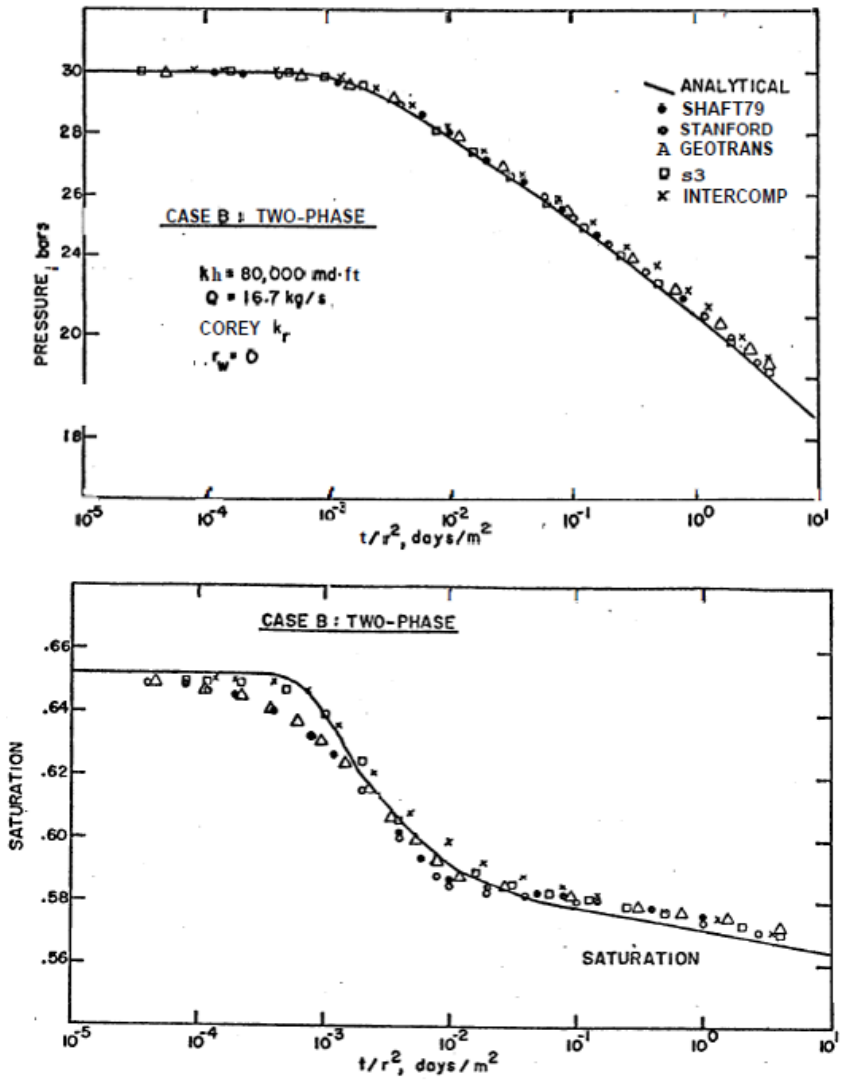

Figure 4.5: Results from the 1980 Stanford Code Comparison Study, Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m)

Moving Flash Front

This simulation involves the propagation of a flash front moving outwards radially from the production well. To verify the results obtained by AD-GPRS, a similar semianalytical result from O'Sullivan (1981) was used. Initially, the reservoir begins

completely filled with liquid water, approximately 5 bars above the saturation pressure for water at a temperature of $573.15K$. This boiling occurs approximately 10 m from the well after about 1 day. Numerically, this problem is more difficult than the previous case due to this phase change behavior. In Figure 4.6, it can be seen that this match is not as close as in the previous cases. The same numerical dispersion was also observed in the smearing of the shock front as seen in the saturation plot. There is also a deviation in the pressure results at the point where the solution goes to two-phase. This same deviation was also observed in the 1980 Code Comparison Study (Figure 4.7).

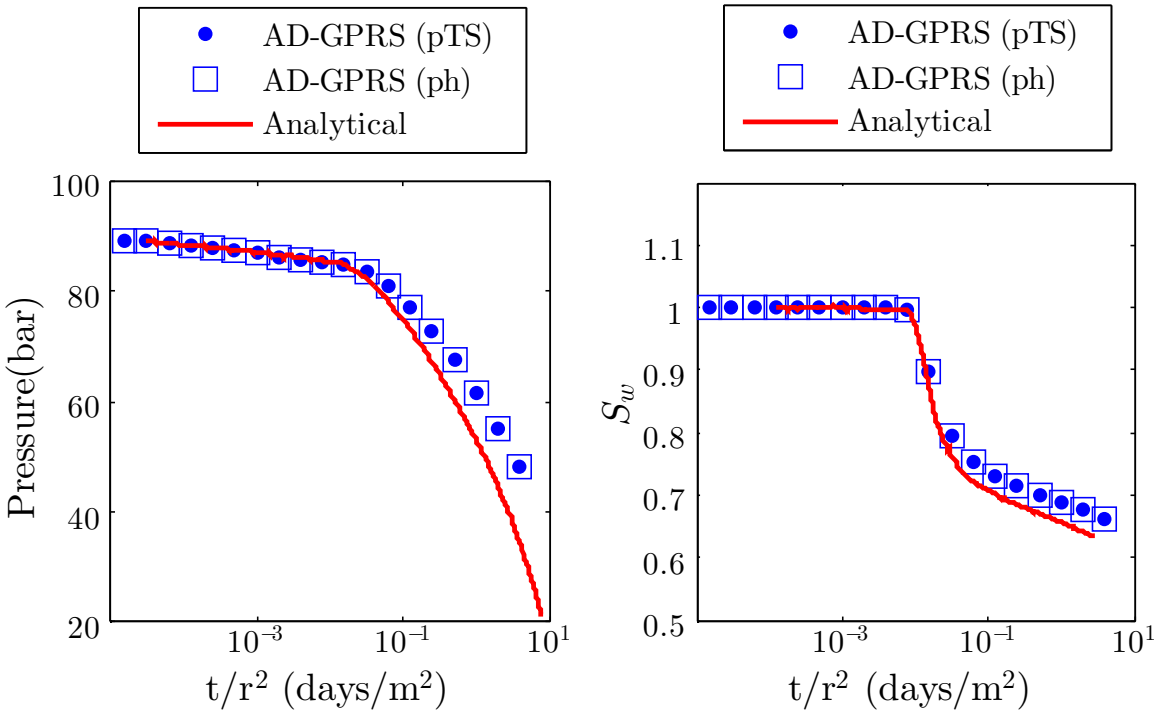

Figure 4.6: Pressure (left) and Water saturation (right) plotted with t/r^2 for the moving flash front case (Case 2C)

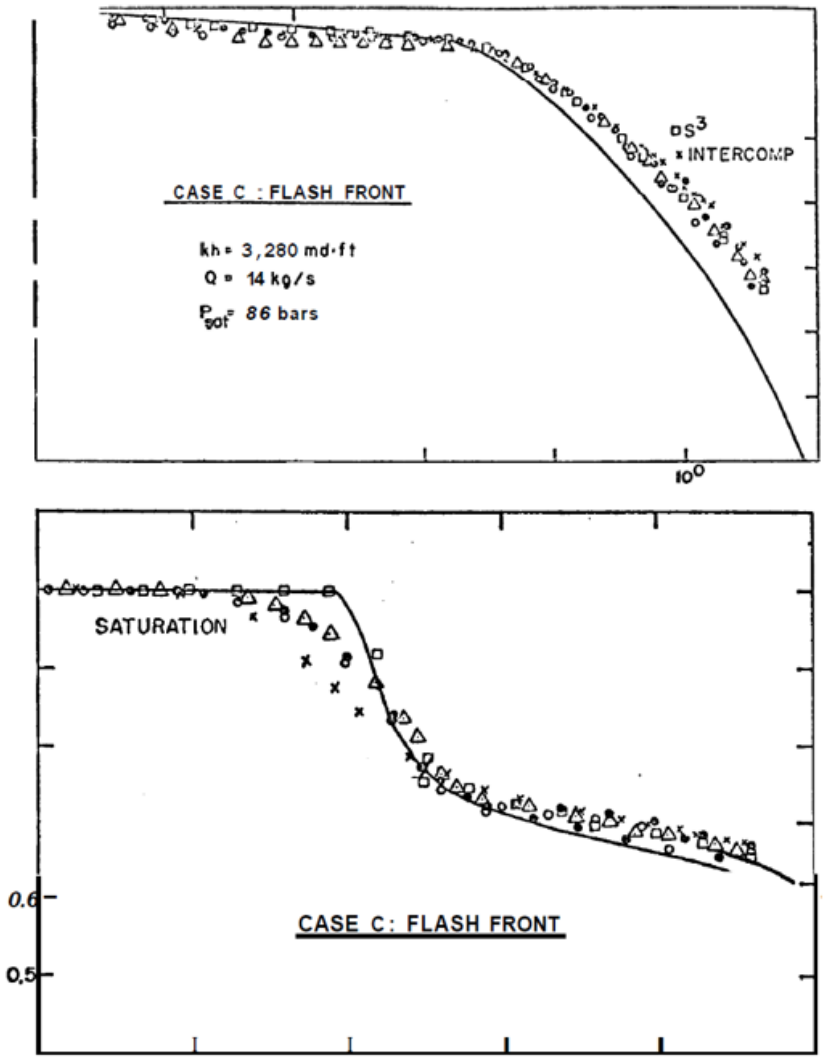

Figure 4.7: Results from the 1980 Stanford Code Comparison Study, Pressure (bar) plotted with respect to t/r^2 (days/m^2) (Plotted at radius of 0.5 m and 0.707 m)(Case 2C)

4.3 Two-phase Boiling with Gravity Drainage

This test case involves a one-dimensional vertical column with both single-phase and two-phase flow. The reservoir is split into two 1 km thick layers with a total of 20

grid blocks. The upper layer is less permeable than the lower layer (exact details are shown in Table 4.3). The initial temperature in the upper layer varies linearly from $283.15K$ to $563.15K$ to the interface and the lower layer varies from $563.15K$ at the interface to $593.15K$ at the bottom of the model. The initial pressure condition in the reservoir reflects a hydrostatic pressure distribution.

The reservoir is being produced at a rate of 100kg/(s km^2) from the bottom block of the model. The grid was chosen to be 20 equally sized blocks and simulated over a 40 year period.

Table 4.3: Model parameters for Test Case 3

	Upper Layer	**Lower Layer**
Rock Density (kg/m^3)	2500	2500
Porosity	0.15	0.25
Permeability (md)	5	100
Rock Specific Heat Capacity (J/gK)	1.0	1.0
Rock Thermal Conductivity (W/mK)	1.0	1.0

The relative permeability relationships used were identical to Case 2 (Equations 4.1, 4.2 and 4.3).

4.3.1 Results

Due to the complexity of this problem, an analytical or semianalytical solution does not exist. Thus, a comparison with the results with TOUGH2, a widely used geothermal simulator was used for the verification instead. Figure 4.8 shows the pressure profiles for three different blocks at decreasing depths. The pressure profiles for each of the blocks have a very close match with the results obtained from TOUGH2. It is important to note that these results are also similarly in agreement with the results obtained in the Code Comparison study (Figures 4.10 and 4.11). From the results

obtained, it is clear that AD-GPRS is able to handle the expansion of the boiling zone in a reservoir and the counter-flow of steam and water. This counter-flow of steam and water due to the gravity effects is evident in real geothermal reservoir fields in Wairakei (Mannington et al., 2004). The saturation (Figure 4.9) results also show a very close match with TOUGH2 over the simulated period. For a two-phase system, having the pressure and saturation to match would fully define the thermodynamic state of the fluid, thus not requiring any other parameters to be matched.

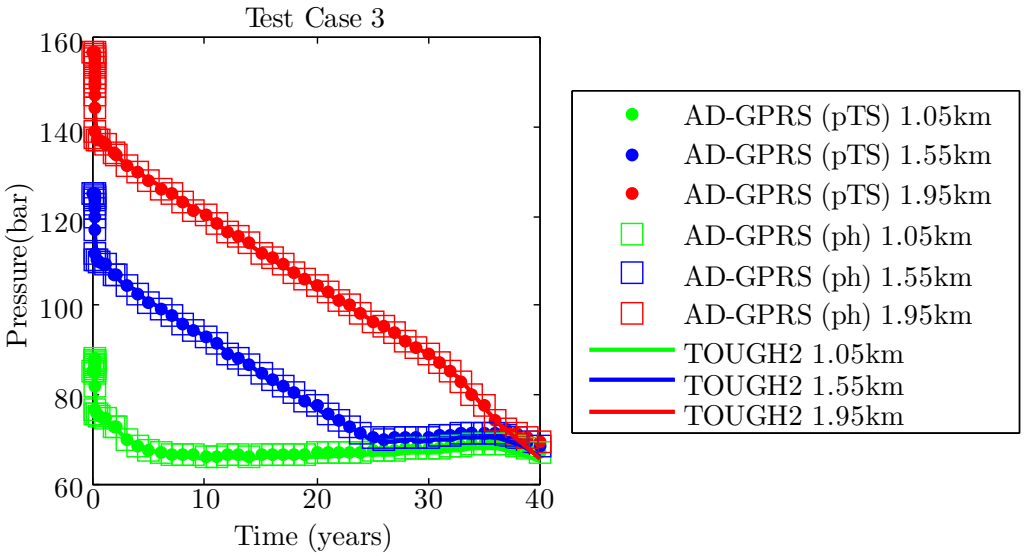

Figure 4.8: Pressure profiles over time for three different blocks at different depths in the models (Case 3)

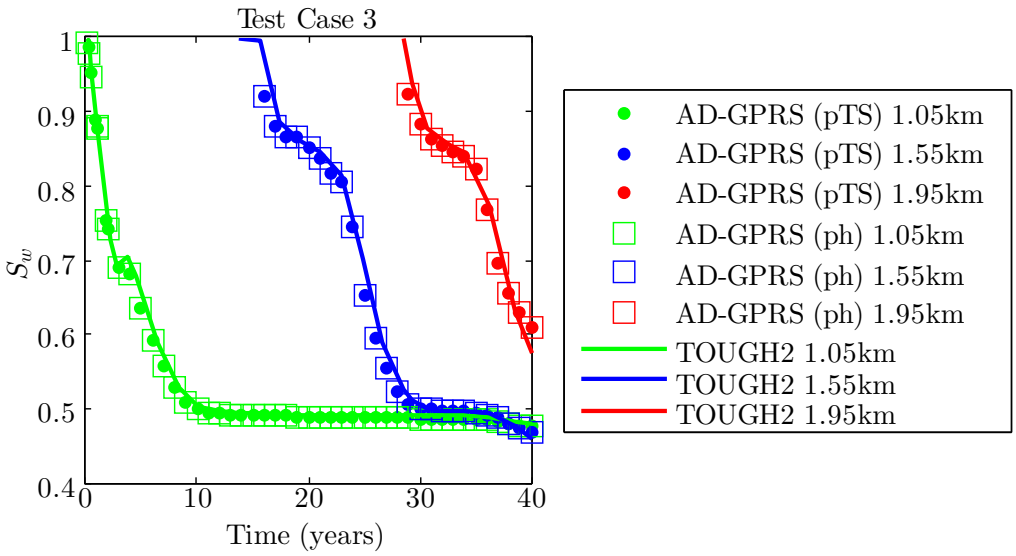

Figure 4.9: Water saturation profiles over time for three different blocks at different depths in the models (Case 3)

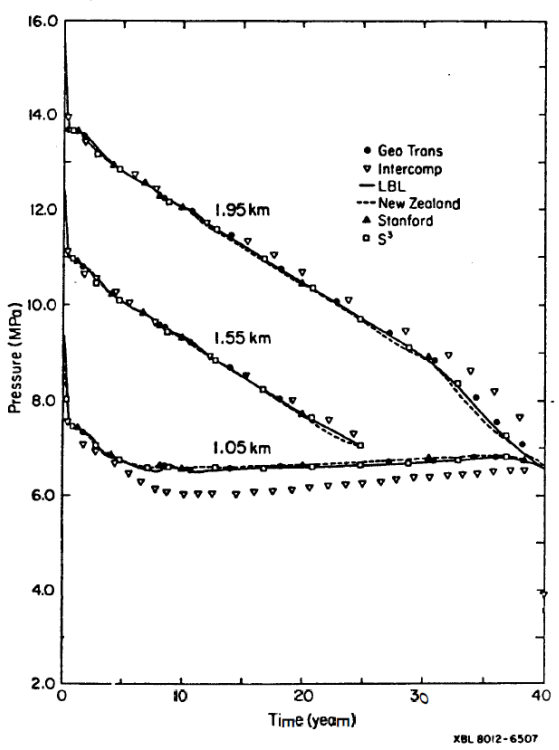

Figure 4.10: Results from the 1980 Stanford Code Comparison Study, Pressure profiles over time for three different blocks at different depths in the models (Case 3)

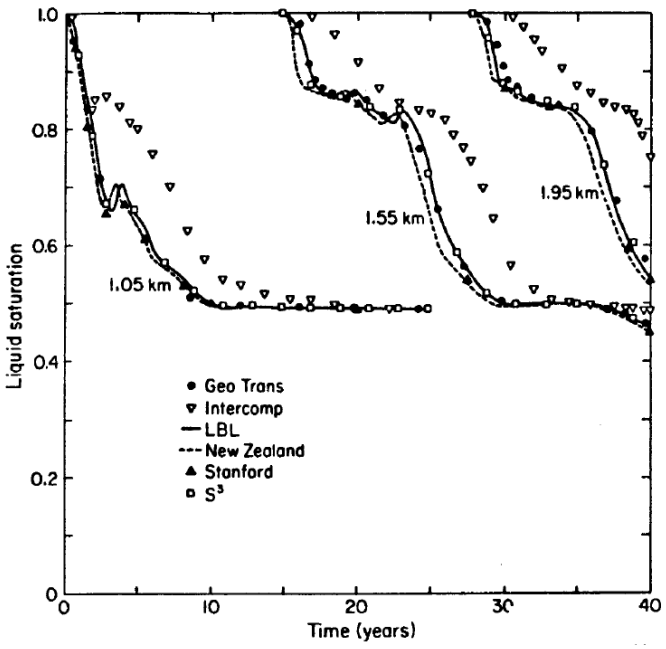

Figure 4.11: Results from the 1980 Stanford Code Comparison Study, Saturation profiles over time for three different blocks at different depths in the models (Case 3)

4.4 Concluding Remarks

The solutions obtained from both the pTS formulation and the ph formulation have been compared for three different test cases. The first test case involved a simple single-phase injection, the second case involved the production from a reservoir at different initial conditions. The solutions from these two test cases were compared with an analytical or semi analytical solution. The results showed a close match. There were small discrepancies due to slightly different thermodynamic relationships used in the semianalytical solution. The final test case of production from a column of fluid with phase change and gravity effects tested the phase change dynamics with gravity and demonstrate a close match. The verification of these two simulators provides confidence that differences in the formulation would be a result in the nonlinear solution process and not a discrepency between the thermodynamic formulations or an error resulting from their implementation.

Chapter 5

Comparison of Formulations

The numerical performance and behavior of the pTS formulation and ph formulation were compared. Two types of test cases were used to examine the numerical behavior of these two formulations. A single block model and a one-dimensional radial model were considered, with both production and injection cases.

The main focus of these comparisons was to examine the behavior of the nonlinear solver as a function of the time step size. In particular, the challenge of phase transitions and how both formulations handle phase transitions was the main focus. These phase transitions are highly nonlinear due to the discontinuity in enthalpy and pressure between the single-phase and two-phase regions. These discontinuities can cause problems for Newton-based nonlinear solvers, where sharp changes in the function values are present. Thus, how the different formulations handle these discontinuities was examined.

A scenario of particular interest involves the of injection of cold water into hot steam. It has been shown in Coats (1980a) that for this particular flow scheme there is significant instability due to the physical phenomenon described as "negative compressibility". This comparison study examined how the pTS and ph formulation perform for this particular flow scheme.

5.1 Single Block Model

The first test case involved a one block model with a single well. Here, a range of initial block conditions was considered for both injection and production scenarios. To compare the nonlinear performance between the two formulations, only a single time step was taken and the performance was quantified by the largest time step that the nonlinear solver could take and still converge to the correct solution. These time steps were calculated by iteratively checking whether a time step converges. The time step recorded is the largest time step at which the nonlinear solver converges. The discrete time step increment was chosen as 0.5 days, so for the Δt reported, the next time step of $\Delta t + 0.5$, will lead to nonconvergent behavior of the nonlinear solver.

The model contained only a single block, with pore volume of 3200m^3, and an initial temperature of 568.3K. This temperature was selected to reflect the high temperature of conventional geothermal reservoirs. A range of different initial conditions for the block was considered. A summary of the initial conditions for each of the different cases can be seen in Table 5.1. (Note: the saturation pressure at a temperature of 568.3K is 80 bar). These initial conditions were chosen to reflect the different typical block conditions that are expected to exist in a geothermal reservoir. These range from liquid water, a two-phase region and a superheated steam region. For the two-phase region, three different saturation cases were tested, a high steam saturation case, high water saturation case and a case with 50% saturated water.

Table 5.1: Initial conditions for the different single block models.

	Initial Conditions		
Case	p[bar]	T[K]	S_s
A	60	568.3	1,0
B	75	568.3	1.0
C	80	568.3	0.99
D	80	568.3	0.5
E	80	568.3	0.01
F	85	568.3	0.0
G	100	568.3	0.0

The linear solver used was GMRES with the BILU preconditioner with a fill-in of 0. The linear solver tolerance was set to 1×10^{-10} for 20 maximum iterations. The nonlinear solver was chosen to have a tight tolerance of 1×10^{-12} and the maximum nonlinear iterations was set to be 20. The relative permeability curves chosen to be X-Curves with zero residual water and steam saturations. Since the well behavior is not of importance, a sufficiently large well index of 10^5 was chosen such that the well constraint would not affect the nonlinear convergence.

Table 5.2: Maximum time step, CFL, number of Newton Iterations and the solution after the corresponding time step for the two formulation in a production scenario (Here the superscript $n+1$ on p, S_s represents the pressure and steam saturation after one time step size of Δt)

	pTS Formulation					**ph Formulation**				
Case	Δt [days]	CFL [-]	p^{n+1} [bar]	S_s^{n+1} [-]	Newton Iters	Δt [days]	CFL [-]	p^{n+1} [bar]	S_s^{n+1} [-]	Newton Iters
A-E	*	*	*	*	*	*	*	*	*	*
F	13.5	0.037	79	0.0560	4	13.5	0.037	79	0.0560	5
G	16	0.044	78	0.0555	4	16	0.044	78	0.0555	5

5.1.1 Production

The production was set to be from a single well that was producing 0.116kg/sec of fluid, the mass of water both in the liquid and gas phase. The initial conditions that were considered are shown in Table 5.1.

Results

The maximum time step was again compared for the two different formulations for a production scenario. The CFL number for the single block case was defined as:

$$CFL_{max} = \frac{\Delta t Q}{PV} \tag{5.1}$$

where Δt is the maximum time step, Q the volume injection rate and PV is the pore volume of the block. For cases A, B, C, D and E, the time step size was limited by the amount of mass in the block rather than the nonlinear solver. This means that if the solution would not converge, it would be because the well would be drawing all the fluid out from the block, resulting in an unphysical solution.

For the conditions that started at liquid water (F, G), both the pTS formulation and ph formulation struggled at crossing this phase transition and only a small time step was able to be taken. For these scenarios, the solutions converged to a solution with only a small ($S_s = 0.05$) steam saturation. This was because in the single-phase liquid region, if the time step taken was too large, the pressure would drop too low and reach a negative pressure. This results in the time step being limited by the pressure dropping to zero in the first Newton iteration. For the production case, we see that the pTS and ph formulation performed equally well for both cases, both in terms of the size of the time step and the number of Newton iterations.

Table 5.3: Maximum time step, CFL, Newton iterations and the solution after the corresponding time step for the two formulations when injecting cold water (Here the superscript $n+1$ on p, S_s represents the pressure and steam saturation after one time step size of Δt)

	pTS Formulation					**ph Formulation**				
Case	Δt	CFL	p^{n+1}	S_s^{n+1}	Newton	Δt	CFL	p^{n+1}	S_s^{n+1}	Newton
	[days]	[-]	[bar]	[-]	Iters	[days]	[-]	[bar]	[-]	Iters
A	2.5	0.059	71	0.9206	4	3	0.071	69	0.9015	9
B	2.5	0.059	72	0.9067	4	1.5	0.036	75.1	0.9456	8
C	25.5	0.606	39	0.0833	9	1.5	0.036	81	0.9440	8
D	∞	*	*	*	*	6	0.143	67	0.2866	4
E-G	∞	*	*	*	*	∞	*	*	*	*

5.1.2 Injection

The injection well was set to be into a single well that was injecting 433.15K water at 1kg/sec. Similar to the production case, the same range of initial conditions were

considered (Table 5.1).

Results

The maximum time step for the different initial conditions for the injection of cold water was analyzed. In Table 5.3, the different maximum time steps (Δt) for the different initial conditions is shown. Again, the CFL number was computed using Equation 5.1, however rather than Q being the produced fluid, Q was the injected fluid instead.

A maximum Δt of ∞ meant that a large enough time step was specified such that fluid could be injected into a block until the point where the pressure becomes unphysical and lies outside of the applicable for the range of the thermodynamic relationships. Thus for these cases (Cases E, F, G) we did not report the time step taken or the time result after that given time step, because the time step is effectively only bounded by the physical limits. For the cases that did have a limit on the time step size, all the cases were not converging due to a Newton iteration converging to a nonphysical solution. This meant that either the p, T or h were outside of the physical or thermodynamic range. This is due to the inaccuracy of the linearization of the nonlinear residual equation in the Newton's Method.

Comparing the pTS and ph Formulation, all the initial condition scenarios examined had similar time step sizes and results, except for cases where the initial conditions of the block are within the two-phase region with a high gas saturation (C($S_s = 0.99$) and D($S_s = 0.5$)). For an initial condition of $S_s = 0.99$, the pTS formulation was able to take about 20 times larger time step size in comparison to the ph formulation. For an initial condition where $S_s = 0.5$ the ph solution still had a limited time step size while the pTS formulation did not.

As expected, the cases where the initial block condition was either fully saturated or had a high steam saturation, both formulations struggled to converge at a large time step. It is important to note that the pTS formulation converges in half the number of Newton iterations. For the cases where the solution started as superheated steam, both the pTS and the ph formulations were only able to converge to a solution at around $S_s \approx 0.9$. The time steps that did not converge, were solutions that converged

to negative pressures that caused the thermodynamic formulations to fail and not continue. This reflects the predicted "negative compressibility" behavior described by (Coats, 1980a), where the pressure would drop while injecting mass. This nonlinear behavior would often cause the nonlinear solution to converge to a negative pressure or enthalpy value.

5.2 One-dimensional Radial Model

The reservoir model for the next comparisons was based on the one-dimensional radial model described in Section 4.2.1. The same reservoir model was used as in Section 4.2.1, however now a more aggressive time stepping scheme was implemented to compare the number of time step cuts between the two formulations.

5.2.1 Production

The total simulation time was set to 10^5 days. The initial time step was chosen to be 1×10^{-5} days with an increment multiplier of 10 and a maximum time step of 10^4 days. The linear solver used was GMRES with the BILU preconditioner with a fill-in of 0. The linear solver tolerance was set to 1×10^{-10} and the maximum iterations was set to 20. The nonlinear solver was chosen to have a tolerance of 1×10^{-6} and the maximum nonlinear iterations was 20.

Results

For this comparison case, the pTS formulation performed better than the ph formulation. The pTS formulation only had five wasted time steps while the ph formulation had 26 wasted time steps. The solutions for these two cases can be seen in Figure 5.1. There is a slight discrepancy between the two solutions if this time stepping strategy was used. This difference was due to the difference in time discretization that the pTS formulation and ph formulation used. This is proved by simulating the same time stepping scheme as used for the ph formulation with a pTS formulation. An exact match in the two simulation results can be seen in Figures 5.2.

Table 5.4: Time step and Newton iteration results for the production from a liquid water reservoir

	PTS	ph
Number time steps	16	37
Total Newton iterations	100	179
Total linear solver iterations	164	322
Wasted time steps	5	26
Wasted Newton iterations	33	119
Wasted linear solver iterations	63	218
CPU time [seconds]	2.7368	6.0136

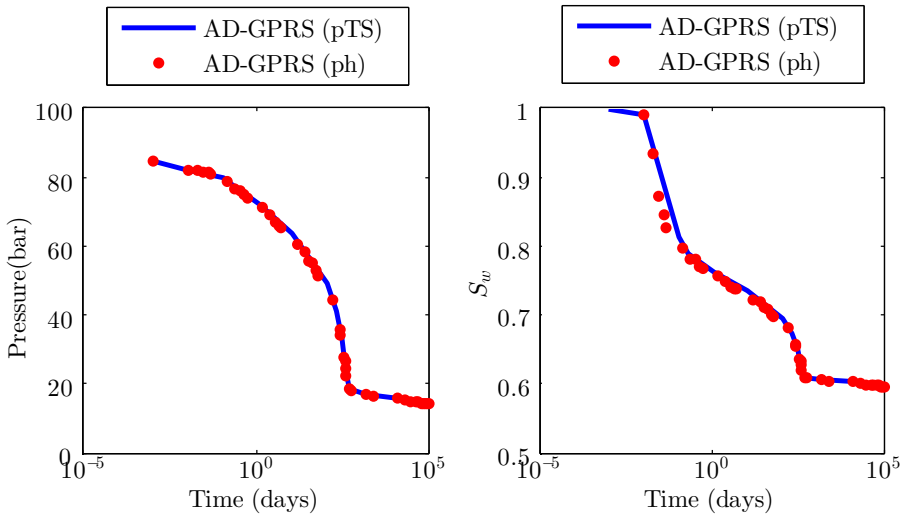

Figure 5.1: Pressure plotted against time (left) and saturation plotted against time (right) for the block at 0.5m away from the wellbore for production from pure liquid water reservoir for the pTS formulation and ph formulation based on the time stepping scheme described in 5.2.1.

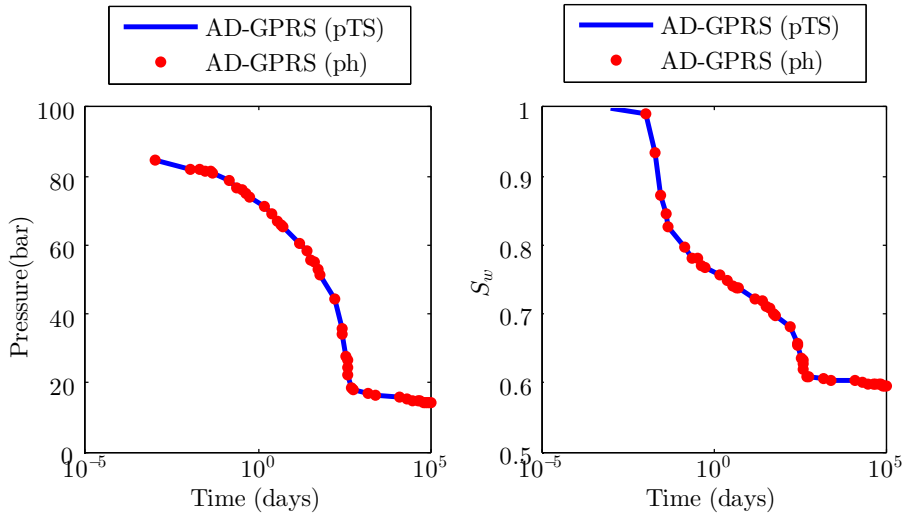

Figure 5.2: Pressure plotted against time (left) and saturation plotted against time (right) for the block at 0.5m away from the wellbore for production from pure liquid water reservoir for thes pTS formulation and ph formulation with both the same time stepping scheme

5.2.2 Injection

The model setup for the injection case follows the model described in Section 4.2.1. However rather than having a producing well, the well is injecting at 10kg/sec at a temperature of 333.15K. Also, the initial condition of the reservoir is 80 bar at a temperature of 573.15K, at superheated steam conditions ($S_s = 1$). The total simulation time was set to be 500 days, with a maximum time step of 100 days. The linear solver used was GMRES and the BILU preconditioner with a fill-in of 0. The linear solver tolerance was set to be 1×10^{-10} for 20 maximum iterations. The nonlinear solver was chosen to have a tolerance of 1×10^5 with the maximum nonlinear iterations to be 100.

Negative Compressibility

This test case is an example of the physical effect described by Coats (1980a) as a "negative compressibility" effect. This phenomenon is a physical reservoir situation

where cold water is injected into a superheated steam region and the cold water causes the steam to condense. This condensation process results in a decrease in the pressure of the system while also increasing the total mass of the system. This effect presents great difficulty for both the nonlinear and linear solver of the simulator.

For the nonlinear solver, due to the linearization of the "compressibility" of the system, this results in instability of the answer that the solution converges to. In particular, a solution may converge to a nonphysical result depending on the specified time step. In Coats (1980a), the analysis was only for the pTS formulation and the impacts of this negative compressibility on a pressure enthalpy formulation are unknown. This section of the study investigated numerically how varying the time stepping scheme could affect the resulting solution for both the pTS and ph formulations.

Time Step Comparison Details

For the two formulations, the results of different time stepping schemes were investigated. These different time stepping schemes involved changing the initial time step and the multiplier for a time step. The initial time step refers to the first time step that the scheme attempts; it is important to note that this time step does not guarantee that it was the first time step scheme, but just the first time step that was attempted. The multiplier refers to the value that a convergent time step is multiplied by to calculate the next time step. Typically in reservoir simulations, a small time step is chosen initially and the time step would eventually increase based on the multiplier after each successive convergent time step. For this study we tested three initial time steps, $10^{-6}, 10^{-4}$ and 10^{-2} days and four different time step multipliers, 1.2, 2, 5 and 10.

Figures 5.4, 5.5, 5.6 5.7 plot the pressure, saturation and temperature solutions, shown radially at the final time step of 500 days. Figures 5.4, 5.5, 5.6, 5.7 correspond to the solutions for different time step multipliers of 1.2, 2, 5 and 10 respectively. Each of the different lines in each of the plots represent a different initial time step. The green, blue and red lines represent the initial time step sizes of $10^{-6}, 10^{-4}$ and 10^{-2} days respectively. The temporal solutions for the block at 0.5m radius can be

seen in the Appendix Figures A.1, A.2, A.3 and A.4.

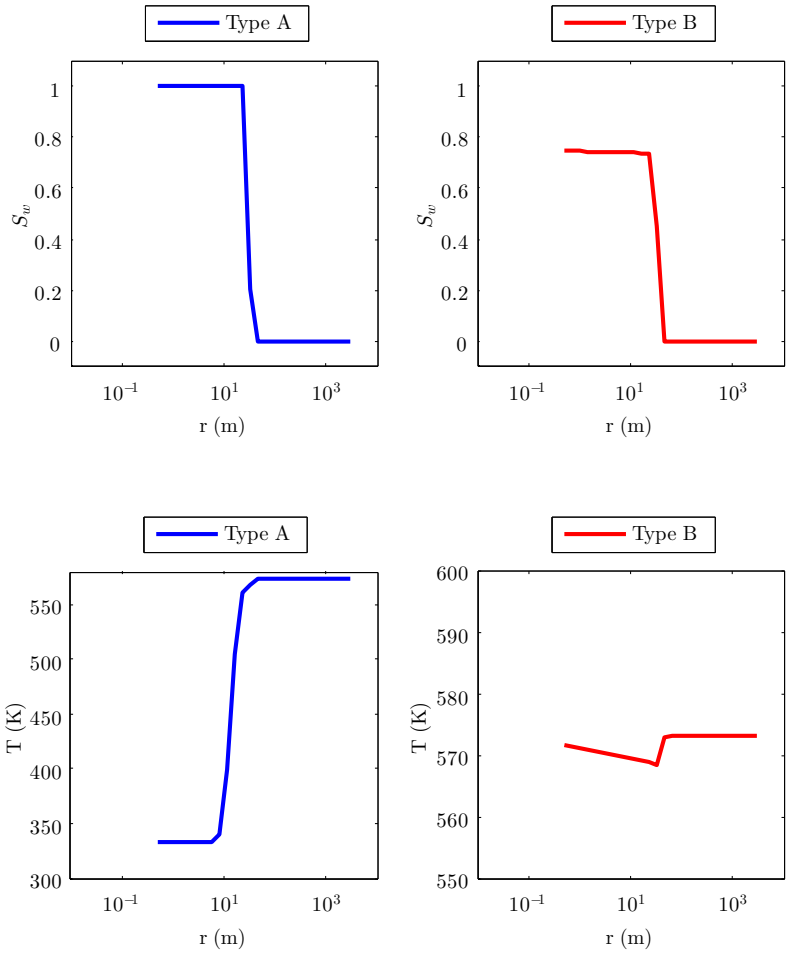

Figure 5.3: Water Saturation (top) and temperature (bottom) solution plotted radially at the time after 500 days as an example of a type A and type B solution

Time Step Comparison Discussion

Depending on the formulation and the time step schemes, there are two main types of solutions that were obtained. The first type of solution, was observed in the pTS formulation solutions at a time step multiplier of 1.2. These solutions are observed to be pure liquid water prior to the phase front, with a very sharp front for the single phase steam region. These types of solutions are referred to as type A solutions.

The other type of solution also involves a sharp front, however the main difference is that before the front the solution is at two-phase conditions. We will refer to these solutions as type B. An example of both of these is shown in Figure 5.3. As shown in the plots, both solutions still are within the physical range of the parameters. The main difference is that the supercooled water part of the type A is replaced by a two-phase solution in type B. This nonuniqueness is a result of the truncation error of different time stepping schemes coupled with the instability of these "negative compressibility" problems.

If a conservative time stepping scheme was used, both the pTS and ph formulations converged to the same type A solution as shown in Figure 5.8. This time stepping scheme has an initial time step of 10^{-4} days, a multiplier of 1.2 and a maximum time step of 0.1 days. We can conclude that because a small step size is more likely to have a more accurate result, this type A solution is most likely the correct solution and the type B solution is the solution resulting from the time truncation error coupled with the unstable nature of "negative compressibility" problems. If larger time steps are taken, both solutions would converge to the type B solution. This is shown by the solutions obtained when the time step multiplier was set to 10. The main difference between these two formulations is the point at which the formulation switches from the type A solution to a type B solution.

At a time step multiplier of 1.2, we can see that for the pTS formulation, (at all three initial time steps) all results converge to type A solutions and are very close to each other. However for the ph formulation, only at a initial time step of 10^{-4} days does the solution converge to the type A solution, while all others converge to type B solutions. For all other time step multipliers, it can be seen that the ph formulation converges only to the type B solutions with slight deviations in the pressure and saturation profiles.

For the pTS formulation, it can be seen that only at a time step multiplier of 10, all the solutions are type B solutions. For time step multipliers of 2 and 5, it can be seen that the solutions converge either to a type A or type B. The overall trend that could be observed was that for an increasing time step multiplier or increasing initial time step size, the solutions tend to converge towards a type B solution. It is

interesting to note that at a time step multiplier of 5 and an initial time step of 10^{-6} for the pTS formulation, it seems as if the solution is a combination of type A and B solution because the temperature prior to the shock is not as low as other type A solutions.

Thus we conclude from these results that the pTS formulation converges to the correct type of solution for larger time steps, while the ph converges to the type B solution. This shows that the pTS formulation's time stepping threshold to converge to the incorrect solution is higher than the ph formulation.

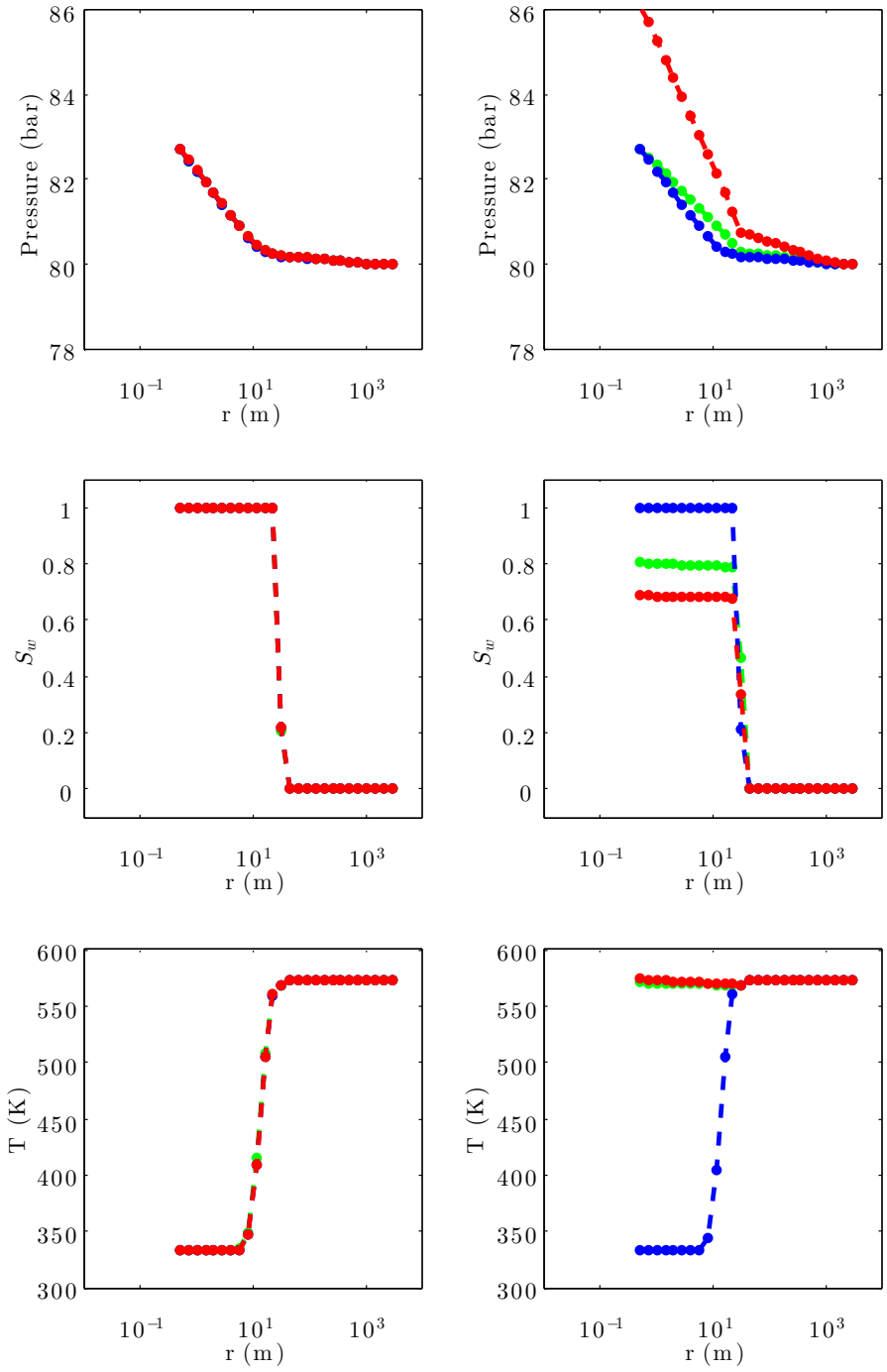

Figure 5.4: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 1.2

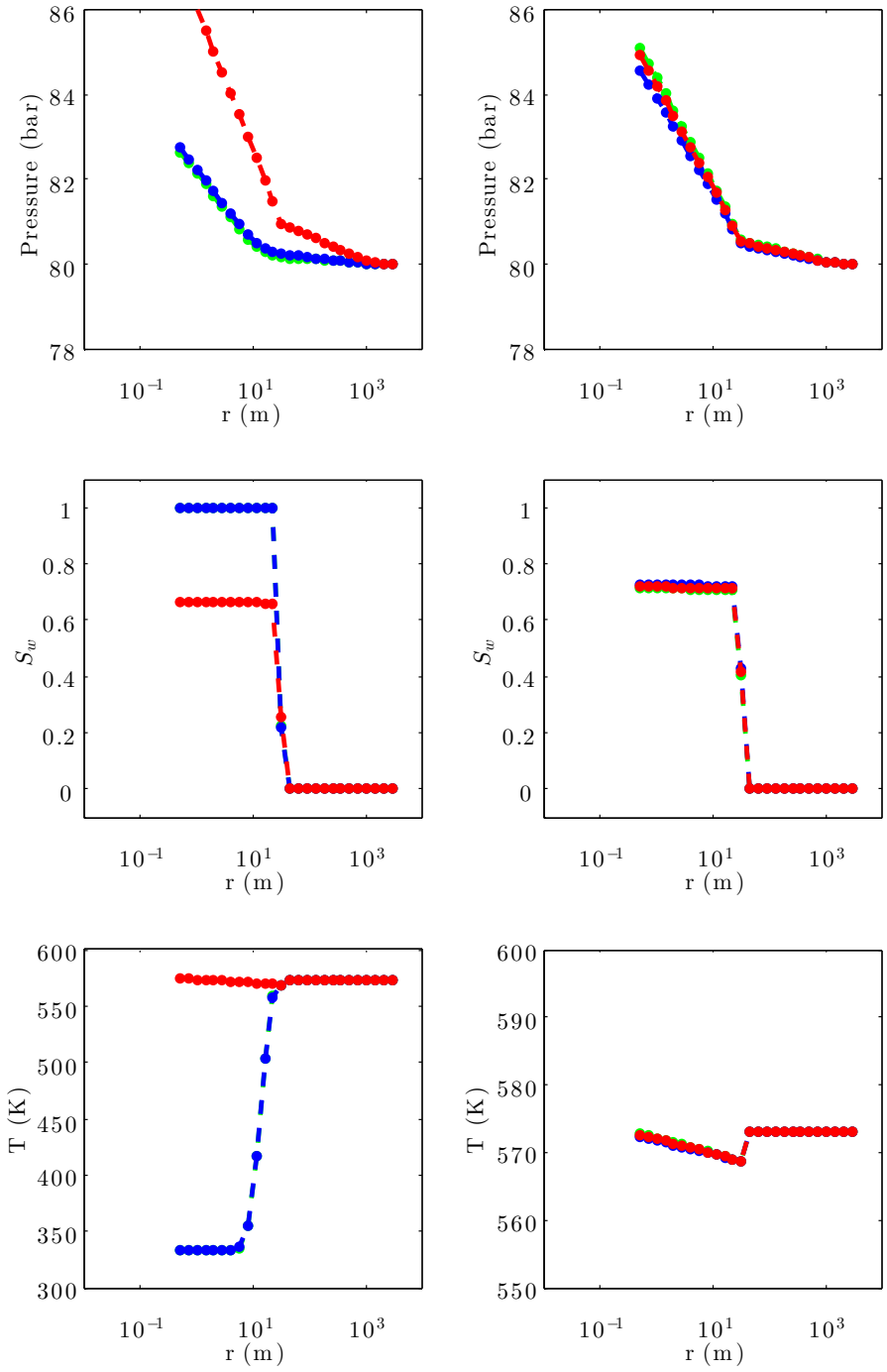

Figure 5.5: Pressure (top) Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 2

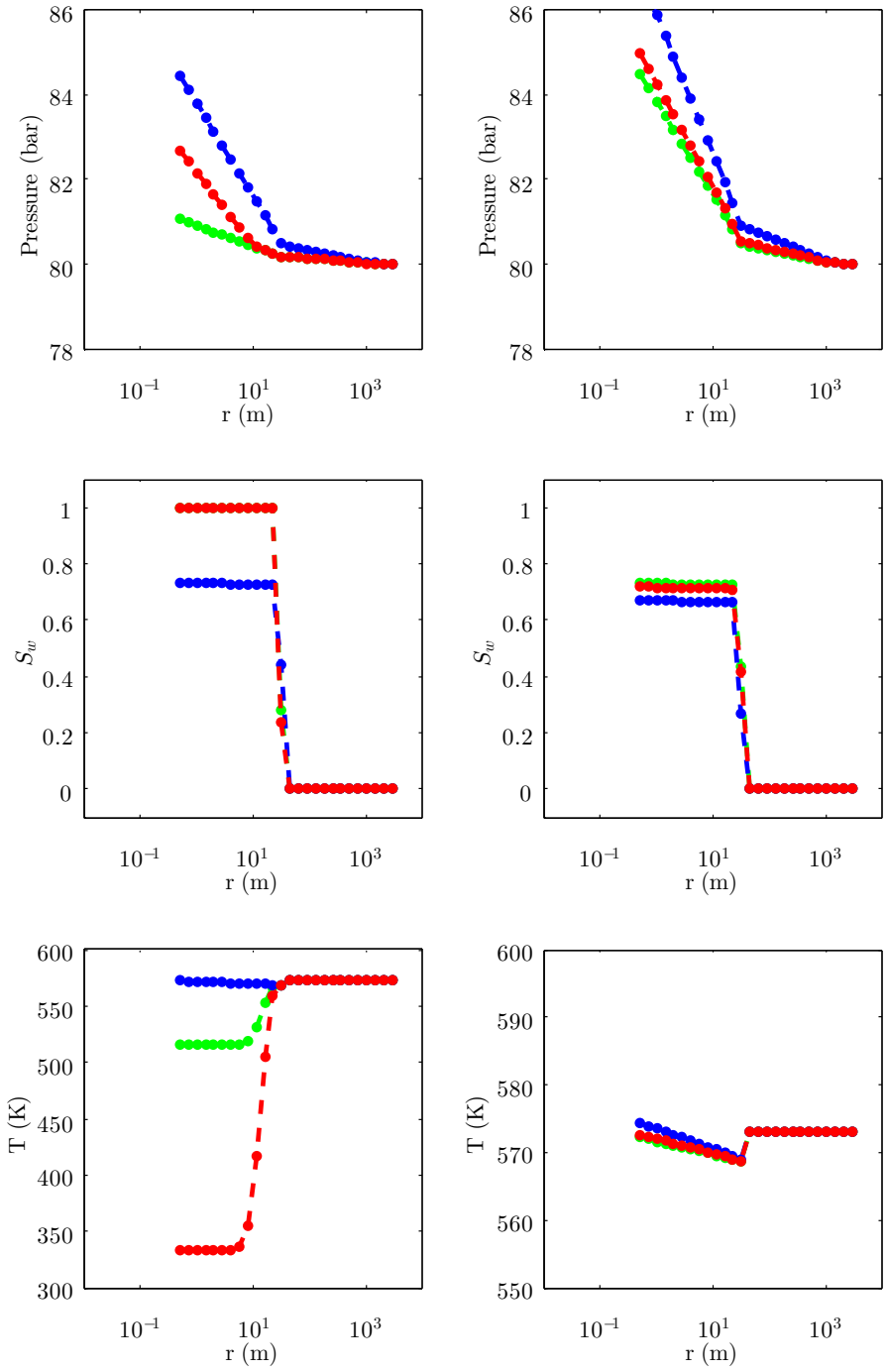

Figure 5.6: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 5

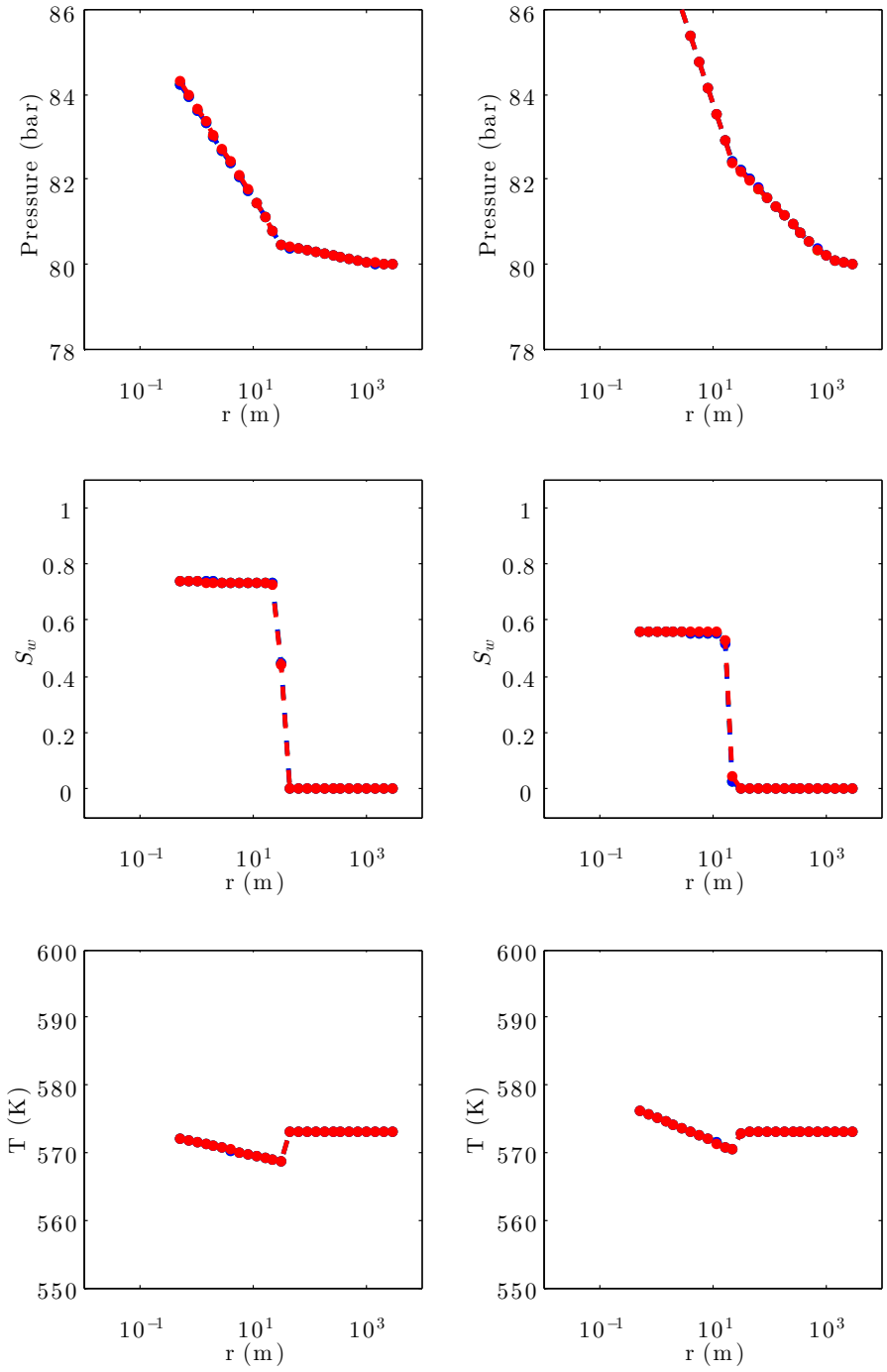

Figure 5.7: Pressure (top), Water Saturation (middle) and Temperature (Bottom) solution plotted radially at the time after 500 days. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 10

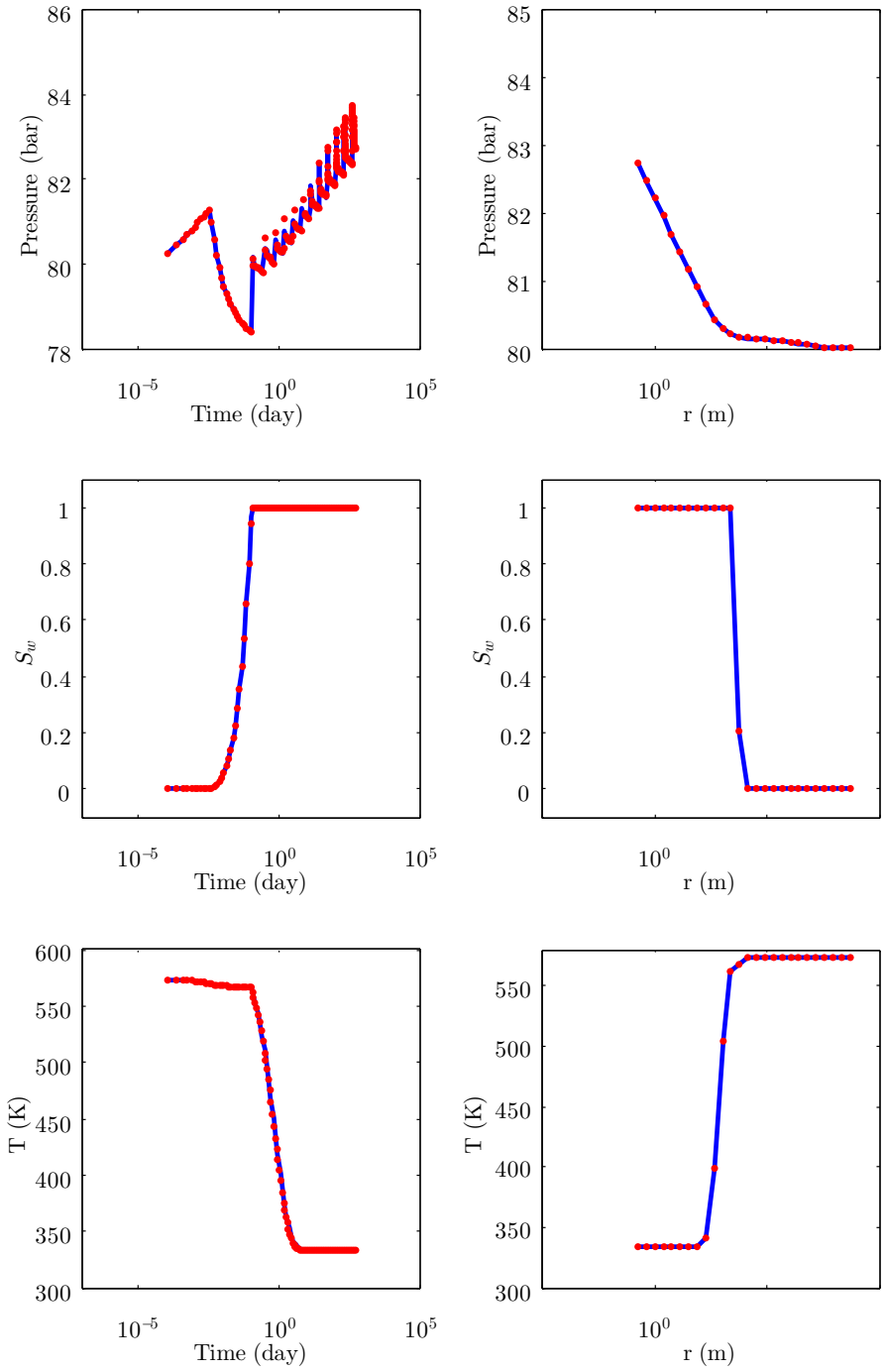

Figure 5.8: Pressure (Top), Water Saturation (middle) and Temperature (bottom) solution plotted against time for the block at 0.5 m radius (left), the solution, plotted against the radius of the block at the final simulation time of 500 days (right). pTS formulation (blue line) and ph formulation (red dots) at initial time steps 10^{-4} time step multiplier of 10 and a maximum time step of 0.1 days.

5.3 Concluding Remarks

A comparison between the pTS formulation and the ph formulation was carried out. For the single block case, it was found that the pTS formulation and ph formulation performed equally well for most initial conditions, production and injection scenarios. For the scenario of injection of cold liquid water into a steam block, the pTS formulation was able to take significantly larger time steps than the ph formulation. For the "negative compressibility" scenario, both formulations had difficulty when injecting cold water into a superheated steam block.

This same trend was reflected in a one-dimensional radial model case. Here, for production from a liquid water reservoir, the pTS performed only slightly better than the ph case. However, injecting cold water into a superheated reservoir caused significant numerical instabilities. It was found that when taking larger time steps, both formulations would converge to a different solution if they were both simulating at a small time step. However this transition between the two types of solution occurs earlier for the ph formulation, which means that for a smaller time stepping scheme, the second solution converges to the type B solution while the pTS formulation would still converge to the type A. Thus, overall it was concluded that the pTS formulation performs better than the ph formulation.

Chapter 6

Conclusions

A geothermal reservoir simulator was developed and implemented with the AD-GPRS framework. The AD-GPRS framework allowed for this implementation of the geothermal module to require only minimal changes. This reduced the amount of code duplication in the implementation. Within this geothermal implementation, this allows for the geomechanics and reaction modelling aspects of AD-GPRS to be applied to future geothermal applications.

This geothermal reservoir simulator supports two formulations, a pressure-temperature-saturation formulation and a pressure-enthalpy formulation. A consistent set of thermodynamic relationships and phase behavior algorithms were developed for these two formulations. This produced identical results on the models tested based on the 1980 Code Comparison Study. The results from both formulations were compared with the analytical, semianalytical and TOUGH2 solutions. For these basic tests they all had good agreement with AD-GPRS, which provided confidence in the solutions obtained from AD-GPRS.

A comparison of the convergence of both nonlinear formulations was investigated. The performance of each nonlinear formulation was quantified based upon the time step sizes and Newton iterations for the different simulations. For the two formulations, the pressure-temperature-saturation formulation performed better overall than the pressure-enthalpy formulation. It was found that when injecting steam into a pure liquid system, there were two solutions that the two formulations converged

to. This was highly dependent on the time stepping scheme used. This dependency was different in the two formulations. The physical solution was computed based on a conservative time stepping scheme. It was found that the pressure temperature and saturation formulation would still converge to this solution for more greedy time stepping schemes while the pressure enthalpy formulation did not.

Further analysis on this time step relation for these "negative compressibility" scenarios could provide a guide as to how to avoid convergence to the incorrect solution. This could be done by analyzing the linearized compressibility in comparison to the nonlinear compressibility. Also, further work could be done only examining how different linear solvers behave for the two formulations and whether the linear solver could be improved in particular for these "negative compressibility" problems.

Appendix A

Additional Plots

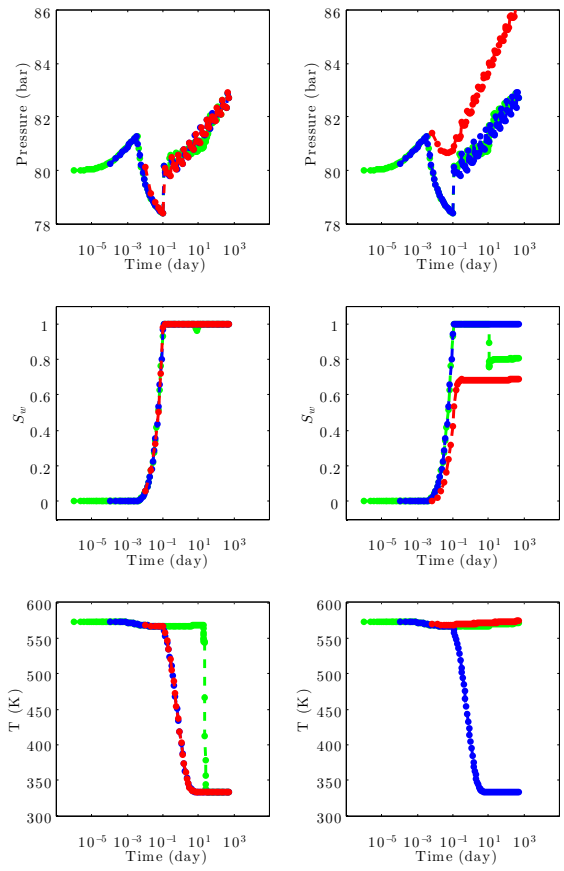

Figure A.1: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 1.2

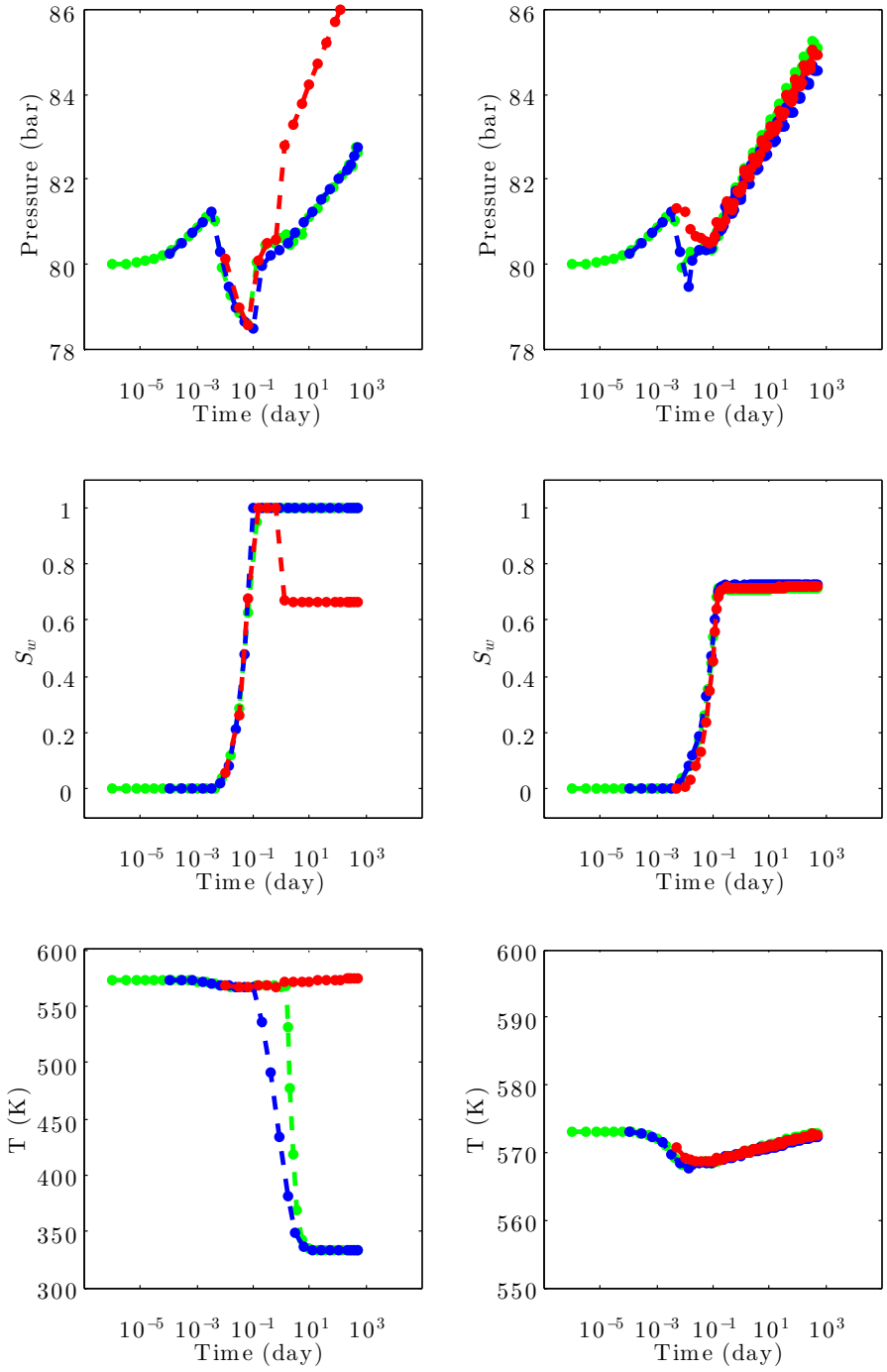

Figure A.2: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 2

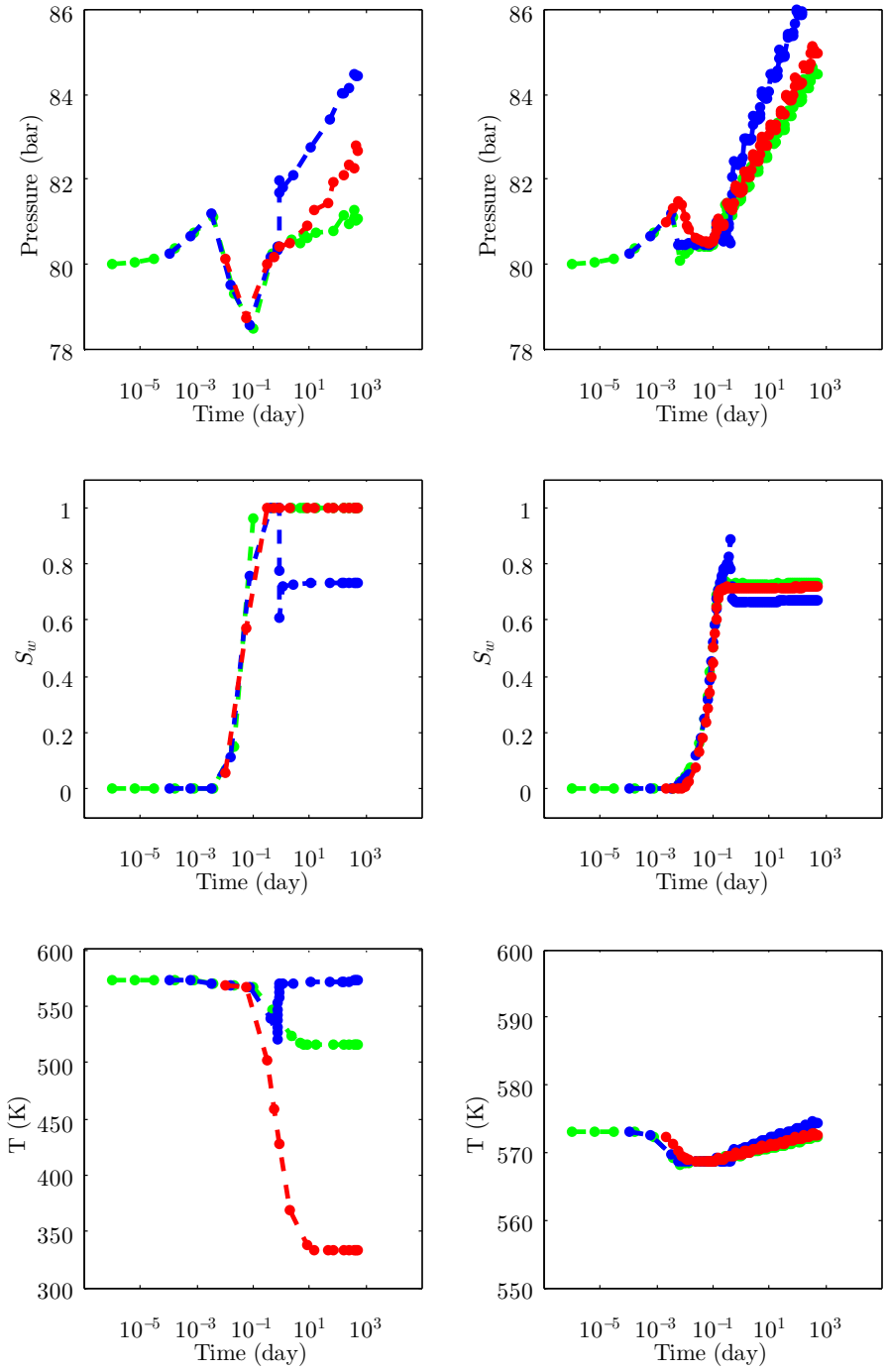

Figure A.3: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 5

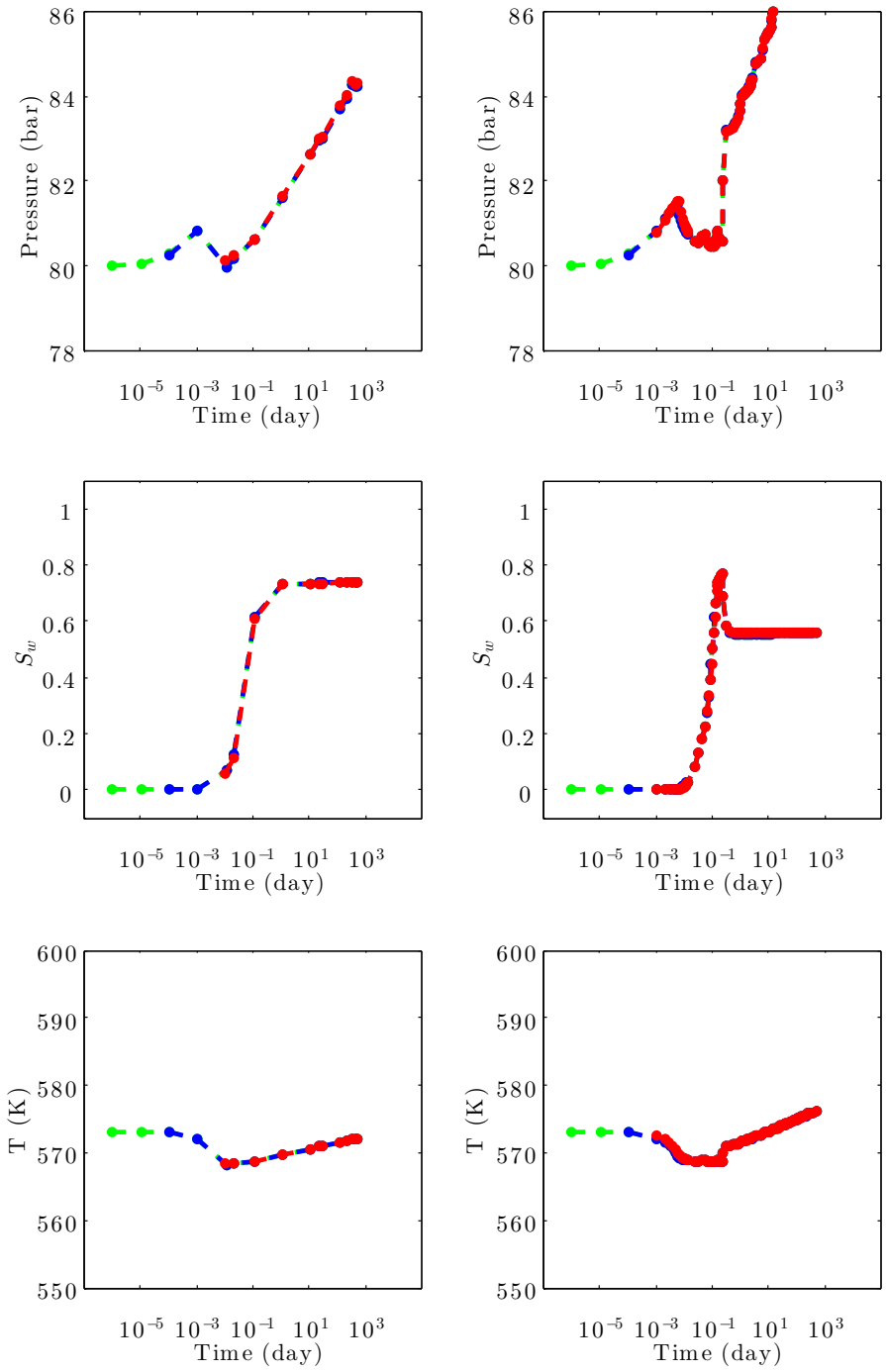

Figure A.4: Pressure (top), Water Saturation (middle) and Temperature (bottom) solution plotted with time for the block at 0.5m. pTS formulation (left) and ph formulation (right) at a range of different initial time steps 10^{-6}(green), 10^{-4} (blue) and 10^{-2} (red) for a time step multiplier of 10

Bibliography

Avdonin, N. (1964). Some formulas for calculating the temperature field of a stratum subject to thermal injection. *Neft'i Gaz*, 3:37–41.

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

Aziz, K. and Settari, A. (1979). *Petroleum reservoir simulation*. Chapman & Hall.

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

Burnell, J., O'Sullivan, M., O'Sullivan, J., Kissling, W., Croucher, A., Pogacnik, J., Pearson, S., Caldwell, G., Ellis, S., Zarrouk, S., and Climo, M. (2015). Geothermal supermodels: the next generation of integrated geophysical, chemical and flow simulation modelling tools. In *Proceedings World Geothermal Congress 2015*.

Cao, H. (2002). *Development of techniques for general purpose simulators*. PhD thesis, Stanford University.

Coats, K. (1980a). Reservoir simulation: A general model formulation and associated physical/numerical sources of instability.

Coats, K. H. (1980b). An equation of state compositional model. *SPE J*, 20(5):363–376.

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

Garipov, T., Levonyan, K., Karimi-Fard, M., and Tchelepi, H. (2012). Coupled geomechanics and flow in fractured porous media. In *ECMOR XIII-13th European Conference on the Mathematics of Oil Recovery*.

GeoQuest, S. (2001). Eclipse technical description manual.

Hernandez, D., Clearwater, J., Burnell, J., Franz, P., Azwar, L., and Marsh, A. (2015). Update on the modeling of the rotokawa geothermal system: 2010–2014. *World Geothermal Congress 2015*.

Iranshahr, A., Voskov, D., and Tchelepi, H. A. (2010). Tie-simplex parameterization for eos-based thermal compositional simulation. *SPE J*, 15(2):545–556.

Kourounis, D., Voskov, D., and Aziz, K. (2010). Adjoint methods for multicomponent flow simulation. In *12th European Conference on the Mathematics of Oil Recovery*.

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

Mannington, W., O'Sullivan, M., and Bullivant, D. (2004). Computer modelling of the wairakei–tauhara geothermal system, new zealand. *Geothermics*, 33(4):401–419.

Noy, D., Holloway, S., Chadwick, R., Williams, J., Hannis, S., and Lahann, R. (2012). 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.

O'Sullivan, J., Croucher, A., Yeh, A., and Mike, O. (2014). Further improvements in the convergence of tough2 simulations. In *11th World Congress on Computational Mechanics*.

O'Sullivan, J., Croucher, A., Yeh, A., and O'Sullivan, M. (2013a). Improved convergence for air-water and co2-water tough2 simulations. In *Proceedings (electronic) 35th New Zealand Geothermal Workshop*, pages 18–20.

O'Sullivan, J., Dempsey, D., Croucher, A., Yeh, A., and O'Sullivan, M. (2013b). Controlling complex geothermal simulations using pytough. In *Proceedings*.

O'Sullivan, M. J. (1981). A similarity method for geothermal well test analysis. *Water Resources Research*, 17(2):390–398.

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

Pritchett, J. (1975). Numerical calculation of multiphase fluid and heat flow in hydrothermal reservoirs. In *Proc. Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA*, pages 201–205.

Pruess, K., Oldenburg, C., and Moridis, G. (1999). Tough2 user's guide version 2. *Lawrence Berkeley National Laboratory*.

Pruess, K., Zerzan, J. M., Schroeder, R. C., and Witherspoon, P. A. (1979). Description of the three-dimensional two-phase simulator shaft78 for use in geothermal reservoir studies. In *1979 Society of Petroleum Engineers of AIME 5th Symposium on Reservoir Simulation, Denver, Colorado*.

Stanford Geothermal Program, S. (1980). Proceedings of the special panel on geothermal model study, report sgp-tr-42. *Energy Resources Engineering, Stanford University*.

Tchelepi, H. and Zhou, Y. (2013). Multi-gpu parallelization of nested factorization for solving large linear systems. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Theis, C. V. (1935). The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using ground-water storage. *Eos, Transactions American Geophysical Union*, 16(2):519–524.

Vanden, K. J. and Orkwis, P. D. (1996). Comparison of numerical and analytical jacobians. *AIAA journal*, 34(6):1125–1129.

Volkov, O. and Voskov, D. V. (2013). Advanced strategies of forward simulation for adjoint-based optimization. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Voskov, D., Zhou, Y., and Volkov, O. (2012). Technical description of ad-gprs. *Energy Resources Engineering, Stanford University*.

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

Voskov, D. V., Tchelepi, H. A., and Younis, R. (2009). General nonlinear solution strategies for multiphase multicomponent eos based simulation. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Yeh, A., Croucher, A., and O'Sullivan, M. J. (2011). Recent experiences with overcoming tough2 memory and speed limits. In *Proceedings: 33rd New Zealand Geothermal Workshop, Auckland, New Zealand*.

Younis, R. M. (2011). *Modern advances in software and solution algorithms for reservoir simulation*. Stanford University.

Zaydullin, R., Voskov, D. V., James, S. C., Henley, H., and Lucia, A. (2014). Fully compositional and thermal reservoir simulation. *Computers & Chemical Engineering*, 63:51–65.

Zhou, Y. (2012). *Parallel general-purpose reservoir simulation with coupled reservoir models and multisegment wells*. PhD thesis, Stanford University.

Zhou, Y., Tchelepi, H. A., and Mallison, B. T. (2011). Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.