TRUST-REGION NEWTON SOLVER FOR MULTIPHASE FLOW
AND TRANSPORT IN POROUS MEDIA

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

Xiaochen Wang
December 2012
I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Dr. Hamdi A. Tchelepi) Principal Adviser

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

(Dr. Louis J. Durlofsky)

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

(Dr. Seong H. Lee)

Approved for the University Committee on Graduate Studies
The simulation of immiscible fluid displacement in subsurface porous media remains an important and challenging problem with application to oil-recovery processes and subsurface CO$_2$ sequestration. The saturation equations, which describe the transport of the fluid phases in space and time, are highly nonlinear. They are characterized by non-convex flux functions that can also be non-monotonic in the presence of strong buoyancy forces. The saturation field is strongly coupled to the phase velocities, which are in turn strongly dependent on the distribution of the phase pressures. In order to model immiscible processes in natural porous media efficiently, it is critical to deal effectively with the nonlinear coupling between flow (pressures and velocities) and transport (saturation).

We describe a new nonlinear solver for immiscible two-phase transport in porous media, where viscous, buoyancy, and capillary forces are significant. The ‘fractional-flow’, $F$, is a nonlinear function of saturation and typically has inflection points and can be non-monotonic. The non-convexity and non-monotonicity of $F$ are major sources of difficulty for the nonlinear solver. A modified Newton algorithm that employs trust-regions of the flux function to guide the Newton iterations is proposed. The flux function is divided into saturation trust regions. The delineation of these regions is dictated by the inflection, unit-flux, and end points. The saturation updates are performed such that two successive iterations cannot cross any trust-region
boundary. If a crossing is detected, we ‘chop back’ the saturation value to the appropriate trust-region boundary. Our trust-region Newton solver has excellent convergence properties across the parameter space of viscous, buoyancy and capillary effects, and it represents a significant generalization of the inflection-point approach of Jenny et al. (JCP, 2009) for viscous dominated flows. We analyze the nonlinear transport equation using low-order finite-volume discretization with phase-based upstream weighting. Then, we prove unconditional convergence of the trust-region Newton method irrespective of the timestep size for single-cell problems. For one-dimensional transport, numerical results across the full range of the parameter space of viscous, gravity and capillary forces indicate that our trust-region scheme is unconditionally convergent. That is, for any choice of the timestep size, the unique discrete saturation solution is found independently of the initial guess. For problems dominated by buoyancy and capillarity, the trust-region Newton solver overcomes the often severe limits on timestep size associated with existing methods. We use complex 3D reservoir models to demonstrate the effectiveness of the proposed trust-region solver. Specifically, we use the top zone of the SPE 10 model (Tarbert formation) and the full SPE 10 model. Compared with state-of-the-art Newton-based nonlinear solvers, our trust-region solver results in superior convergence performance, and it reduces in the total Newton iterations by more than an order of magnitude, which leads to a comparable reduction in the overall computational cost.

We then describe a nonlinear solution algorithm for coupled flow and transport in heterogeneous porous media where both the viscous and buoyancy forces are significant. We show that flow reversals between Newton updates (or timesteps) are the primary source of nonlinear convergence problems for coupled multiphase flow and transport. For a given flux function, the combinations of saturation and phase-flow direction that lead to flow-reversal are enumerated. These ‘flip points’, which can be computed a-priori, identify the interfaces that experience a flow reversal as the
saturation fields evolves over the current iteration, or timestep. This flow-reversal information is used to update the flow field to ensure consistency of the residual equations of both flow and transport and the associated Jacobian. If flow reversal is detected anywhere in the model for a given timestep, we switch from the sequential-implicit method (SIM) to the fully-implicit method (FIM) using the latest estimates of pressure and saturation as initial guesses. Numerical evidence shows that the proposed nonlinear solver is able to converge for timesteps that are much larger than what the SIM can handle. In addition, for very large timestep sizes, the new nonlinear solver yields better convergence performance than standard FIM.

A preconditioning strategy to overcome convergence difficulties in the nonlinear solver that are associated with the propagation of saturation fronts into regions that are at, or near, the residual saturation is proposed. The convergence difficulties are due to unphysical mass accumulation in certain grid blocks during the Newton iterations. The unphysical mass accumulation, which is proportional to the throughput over the timestep, propagates in the computational domain quite slowly with Newton iterations - as slow as one grid block per iteration. As a result, convergence is often not possible, and when it occurs, it can be quite slow, especially for large throughput (i.e., large timesteps). We propose a strategy that guarantees that mass in any grid block moves no slower than in any of its upwind grid blocks. The preconditioning strategy leads to monotonic iterative updates of the saturation field, resulting in rapid convergence. Numerical examples show that this strategy accelerates the convergence of existing nonlinear solvers quite significantly, especially for aggressively large timesteps.
I would like to express my utmost gratitude towards my advisor, Prof. Hamdi Tchelepi. I have been working with Hamdi for more than five years, and I sincerely think that it is a great fortune to have him as my advisor. I appreciate his advice, guidance, and cutting-edge vision on my research work. More importantly, the encouragement and support from Hamdi gave me a lot of confidence in my research work. He provided me with the big picture and motivation of the research work when I started, pointed out specific directions to pursue when I got confused, and encouraged me to overcome the difficulties when I felt frustrated.

Sincere thanks to Professor Lou Durlofsky, who read my thesis carefully and gave insightful comments and constructive feedback on revising and improving my thesis.

I would also like to thank Dr. Seong Lee for taking me as an intern for two summers. It was a pleasure working with him. With his broad knowledge and deep understanding in reservoir simulation, Seong always provides valuable suggestions for my work. Moreover, I appreciate his patience and time for serving in the reading committee.

Many thanks to Dr. Denis Voskov and Dr. Rami Younis for their help and advice in my research work. Special appreciation goes to Dr. Hui Zhou and Dr. Yaqing Fan. This research would not have been possible without their constant input and moral support.

Needless to say, I am grateful to all of my colleagues at Stanford. I would like to
express my special appreciation to Jingyi Chen, Yunyue Li, Yao Tong and Yi Shen for their company during the past years. Having those close friends has been indeed a gift for me through these years - I deeply treasure the heart-warming and sweet time we spent together. I am especially thankful to Monrawee Pancharoen for being my officemate for five years and sharing my joys and frustrations. I also cherish the friendship with other great friends: Christin Strandli, Panithita Rochana, Chia-Wei Kuo, Zhe Wang, Yixuan Wang, Hangyu Li, Lin Zuo, Jingcong He, Boxiao Li, Da Huo, Alireza Iranshahr, Hadi Hajibeygi, Mohammad Shahvali, Karine Levonyan, Sara Farshidi, Carla Co, Matthieu Rousset, Obi Isebor, and Abby Kirchofer.

I am indebted to my parents, who provide me unconditional love and support for the past 27 years. I thank my husband, Yifan Zhou, for his company, support and love throughout the journey. He was, is, and will be the most important person in my life.
Contents

Abstract iii
Acknowledgments vi

1 Introduction 1
  1.1 Mathematical model ....................................... 2
  1.2 Challenges for nonlinear solvers ......................... 7
  1.3 Thesis outline ........................................... 12

2 Nonlinear Solvers: State of the Art 13
  2.1 Modified Newton method ................................ 14
  2.2 Reduced-Newton method ................................ 16
  2.3 Continuation-Newton method .......................... 19
  2.4 Ordinary trust-region method: NLEQ-RES algorithm .... 21

3 Trust-Region Newton for Transport 25
  3.1 Discretized transport problem ......................... 26
  3.2 Cocurrent flow ........................................... 30
  3.3 Counter-current flow due to gravity ................ 32
  3.4 Capillarity ................................................ 35
  3.5 General transport scheme ............................... 37
3.6 Global convergence of multiphase transport  ........................................ 39
3.7 Numerical examples .................................................................................. 41
  3.7.1 Single-cell problems: convergence maps ........................................... 42
  3.7.2 Two-cells transport: Newton flow ....................................................... 56
  3.7.3 1D transport ......................................................................................... 62
  3.7.4 Comparison with NLEQ-RES algorithm ............................................. 79
  3.7.5 Large heterogeneous examples ............................................................. 87

4 Trust-Region Newton for Flow and Transport  ........................................ 96
  4.1 Flow reversal ........................................................................................... 98
  4.2 SIM→FIM .................................................................................................. 106
  4.3 Numerical examples .................................................................................. 108
    4.3.1 Simple 3×3 example ......................................................................... 108
    4.3.2 2D examples ....................................................................................... 111

5 Preconditioning for Nonlinear Solvers .................................................... 122
  5.1 Linear relative permeability curves ......................................................... 126
  5.2 More general relative permeability curves .............................................. 129
  5.3 Preconditioning strategy ......................................................................... 134
  5.4 Numerical examples .................................................................................. 137
    5.4.1 2D flow and transport ...................................................................... 138
    5.4.2 3D flow and transport ...................................................................... 142

6 Conclusions and Future Work .................................................................. 147
  6.1 Summary and conclusions ....................................................................... 147
  6.2 Recommendations for future work ........................................................... 150

A Asymptotic Limit of Capillary Diffusion Coefficient ................................ 154
  A.1 Brooks-Corey model ............................................................................... 155
List of Tables

3.1 Comparison for iterative solution of a two-cell problem with viscous forces only ........................................ 61
3.2 Comparison for iterative solution of a two-cell problem with viscous forces and buoyancy ........................................ 64
3.3 Performance comparison between NLEQ-RES and TR for 1D transport problem with viscous forces ............................. 85
3.4 Performance comparison between NLEQ-RES and TR for 1D gravity segregation ........................................ 86
3.5 Summary of performance for upper zone of SPE 10 model .................................................................................. 89
3.6 Summary of performance for SPE 10 model ....................................................................................................... 94
4.1 Performance comparison for a 2D incompressible two-phase gravity segregation problem (wetting phase on top and nonwetting phase at bottom) ........................................ 114
4.2 Performance comparison for a 2D incompressible two-phase gravity segregation problem (wetting at left and nonwetting at right) ........................................ 117
4.3 Performance for a 2D incompressible two-phase quarter five-spot problem without gravity ........................................ 119
4.4 Performance comparison for a 2D incompressible two-phase quarter five-spot problem with gravity ........................................ 121
5.1 Summary of performance for two phase flow and transport in bottom layer of SPE 10 model (60 × 220 = 13,200), in the presence of buoyancy forces. End of simulation: \( T = 100 \) days. Timestep size: \( \Delta t = 100 \).

5.2 Summary of performance for two phase flow and transport in top 10 layers of SPE 10 model (60 × 220 × 10 = 132,000), in the presence of buoyancy forces. End of simulation: \( T = 100 \) days. Timestep size: \( \Delta t = 100 \).

D.1 Analytical segregation time for gravity segregation with linear relative permeability curves.

D.2 Nonlinear performance of gravity segregation with linear relative permeability curves.

D.3 Analytical segregation time for gravity segregation with quadratic relative permeability curves.

D.4 Nonlinear performance of gravity segregation with quadratic relative permeability curves.

D.5 Nonlinear performance of displacement under viscous and buoyancy forces with linear relative permeability curves.

D.6 Nonlinear performance of displacement under viscous and buoyancy forces with quadratic relative permeability curves.

F.1 Performance of preconditioning (number of iterations) for 1D transport under viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \).

F.2 Performance of preconditioning (number of iterations) for 1D transport under viscous forces. \( k_{rw} = S^3 \) and \( k_{rn} = (1 - S)^3 \).

F.3 Performance of preconditioning (number of iterations) for 1D transport under viscous forces. \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^2 \).
F.4 Performance of preconditioning (number of iterations) for 1D transport under viscous forces. $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^{10}$.  

F.5 Performance of preconditioning (number of iterations) for 1D transport under viscous and buoyancy forces. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. $N_g = -5$.  

F.6 Performance of preconditioning (number of iterations) for 1D transport under viscous and buoyancy forces. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $N_g = -5$.  

F.7 Performance of preconditioning (number of iterations) for 1D gravity segregation. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$.  

F.8 Performance of preconditioning (number of iterations) for 1D gravity segregation. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$.  

F.9 Performance comparison for transport in SPE 10 top layer. $T = 10$ days.  

F.10 Performance comparison for transport in SPE 10 top layer. $T = 100$ days.  

F.11 Performance comparison for transport in SPE 10 bottom layer. $T = 10$ days.  

F.12 Performance comparison for transport in SPE 10 bottom layer. $T = 100$ days.
List of Figures

1.1 S-shape flux function for two-phase flow ........................................... 6
1.2 Illustrative triangle representing the scope of this work ......................... 9

2.1 Flow chart for modified Newton method in Jenny et al. (2009) .................. 15
2.2 Gas saturation for immiscible gas displacement ..................................... 16
2.3 Flow chart for the reduced-Newton method in Kwok and Tchelepi (2007) .... 18
2.4 Illustration of solution path and iterative solutions for Newton method ....... 20
2.5 Illustration of iterative solutions using the continuation-Newton approach (from Younis et al. (2010)) ........................................ 21

3.1 Illustration of the flux between two neighboring cells ............................... 26
3.2 Convergence ratio with viscous flux ..................................................... 31
3.3 Convergence ratio with viscous and gravitational flux ............................. 33
3.4 Convergence ratio with viscous and capillary flux .................................. 36
3.5 Flow chart of trust-region Newton method ............................................ 38
3.6 Illustration for single-cell problem ....................................................... 42
3.7 Standard Newton scheme: convergence map for viscous flux .................. 43
3.8 Trust-region Newton scheme: convergence map for viscous flux ............... 44
3.9 Standard Newton scheme: convergence map for viscous and gravitational flux ($N_g = -5$) .................................................. 46
3.10 Trust-region Newton scheme: convergence map for viscous and gravitational flux ($N_g = -5$) ........................................... 47
3.11 Newton scheme with Appleyard chopping: convergence map for viscous and gravitational flux ($N_g = -5$) ......................... 48
3.12 Standard Newton scheme: convergence map for viscous and gravitational flux ($N_g = -20$) ........................................ 50
3.13 Trust-region Newton scheme: convergence map for viscous and gravitational flux ($N_g = -20$) ........................................ 51
3.14 Newton scheme with Appleyard chopping: convergence map for viscous and gravitational flux ($N_g = -20$) ......................... 52
3.15 Standard Newton scheme: convergence map for viscous and capillary flux ................................................................. 54
3.16 Trust-region Newton scheme: convergence map for viscous and capillary flux ................................................................. 55
3.17 Standard Newton scheme: convergence map for viscous, gravitational and capillary flux ................................................... 57
3.18 Trust-region Newton scheme: convergence map for viscous, gravitational and capillary flux ................................................... 58
3.19 Illustration of two-cell problem ................................................. 58
3.20 Newton flow for viscous flux .................................................... 60
3.21 Newton flow for viscous and gravitational flux ............................ 63
3.22 Numerical results with trust-region Newton scheme of Buckley-leverett displacing problem under viscous forces ......................... 66
3.23 Performance comparison of a Buckley-leverett displacing problem under viscous forces ....................................................... 67
3.24 Numerical results with trust-region Newton scheme of a 1D gravity segregation problem ..................................................... 69
3.25 Performance comparison of a 1D gravity segregation problem  
3.26 Numerical results with trust-region Newton scheme of a 1D heterogeneous capillarity segregation problem  
3.27 Performance comparison of a 1D heterogeneous capillarity segregation problem  
3.28 Numerical results with trust-region Newton scheme of a Buckley–leverett displacing problem under viscous and buoyancy forces  
3.29 Performance comparison of a Buckley–leverett displacing problem under viscous and buoyancy forces  
3.30 Numerical results with trust-region Newton scheme of a 1D gravity and (homogeneous) capillarity segregation problem  
3.31 Performance comparison of a 1D gravity and (homogeneous) capillarity segregation problem  
3.32 Numerical results with trust-region Newton scheme of a Buckley–leverett displacing problem under viscous, buoyancy and capillary forces  
3.33 Performance comparison of a Buckley–leverett displacing problem under viscous, buoyancy and capillary forces  
3.34 Newton flow for viscous flux  
3.35 Newton flow for viscous and gravitational flux  
3.36 Numerical results of a 1D gravity segregation problem  
3.37 Relative permeability curves in large test case  
3.38 CFL distribution for upper zone of SPE 10 model (Tarbert formation)  
3.39 Comparison of total Newton iterations for upper zone of SPE 10 model (Tarbert formation)  
3.40 Comparison of total simulation time for upper zone of SPE 10 model (Tarbert formation)
3.41 Comparison of Newton iterations per timestep for upper zone of SPE 10 model (Tarbert formation) .......................................................... 92
3.42 Comparison of total Newton iterations for SPE 10 model .......................... 93
3.43 Comparison of total simulation time for SPE 10 model .............................. 93
3.44 History of the maximum CFL number for SPE 10 model ................................ 95

4.1 Flux functions for two-phase flow in the presence of strong buoyancy 100
4.2 Four types of flow reversal under strong buoyancy .................................... 101
4.3 Flow chart of SIM→FIM .......................................................... 107
4.4 Numerical results with SIM→FIM for a 3*3 two-phase gravity segregation problem ................................................................. 110
4.5 Permeability distribution of 2D examples .................................................. 112
4.6 Numerical results with SIM→FIM for 2D two-phase gravity segregation problem (Initially wetting on top and nonwetting at bottom) .. 113
4.7 Numerical results with SIM→FIM: 2D two-phase gravity segregation problem (Initially wetting at left and nonwetting at right) .... 116
4.8 Numerical results with SIM→FIM: 2D two-phase quarter five-spot problem without gravity .......................................................... 118
4.9 Numerical results with SIM→FIM: 2D two-phase quarter five-spot problem with gravity .......................................................... 120

5.1 Solutions for 1D transport problem with only viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). ........................................ 124
5.2 Spikes in solutions for 1D transport problem with only viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \) ........................................ 124
5.3 S-shape flux function for two-phase flow .................................................. 125
5.4 Solutions of trust-region Newton method for 1D transports with viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). ........................................ 136
5.5 Permeability of SPE 10 bottom layer

5.6 CFL distribution for two phase flow and transport in bottom layer of SPE 10 model \((60 \times 220 = 13,200)\), in the presence of buoyancy forces.

End of simulation: \(T = 100\) days. Timestep size: \(\Delta t = 100\).

5.7 Total number of (FIM) Newton iterations for two phase flow and transport with viscous forces in bottom layer of SPE 10 model \((60 \times 220 \times 10 = 132,000)\). End of simulation: \(T = 1000\) days. Timestep size: \(\Delta t = 1000\).

5.8 Total simulation time for two phase flow and transport with viscous forces in bottom layer of SPE 10 model \((60 \times 220 \times 10 = 132,000)\).

End of simulation: \(T = 1000\) days. Timestep size: \(\Delta t = 1000\).

5.9 CFL distribution for two phase flow and transport in top 10 layers of SPE 10 model \((60 \times 220 \times 10 = 132,000)\), in the presence of buoyancy forces. End of simulation: \(T = 100\) days. Timestep size: \(\Delta t = 100\).

5.10 Total number of (FIM) Newton iterations for two phase flow and transport in top 10 layers of SPE 10 model \((60 \times 220 \times 10 = 132,000)\), in the presence of buoyancy forces. End of simulation: \(T = 100\) days.

5.11 Total simulation time for two phase flow and transport in top 10 layers of SPE 10 model \((60 \times 220 \times 10 = 132,000)\), in the presence of buoyancy forces. End of simulation: \(T = 100\) days. Timestep size: \(\Delta t = 100\).

A.1 Asymptotic limit of capillary diffusion coefficient: Brooks-Corey model

A.2 Asymptotic limit of capillary diffusion coefficient: van Genuchten model

B.1 Numerical solution for segregation under (homogeneous) capillarity

B.2 Wetting-phase fluxes: segregation with (homogeneous) capillary forces
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.3 Numerical solution for segregation under (heterogeneous) capillarity</td>
<td>166</td>
</tr>
<tr>
<td>B.4 Wetting-phase fluxes: segregation under (heterogeneous) capillarity</td>
<td>167</td>
</tr>
<tr>
<td>D.1 Flux functions of gravity segregation</td>
<td>171</td>
</tr>
<tr>
<td>D.2 Initial condition of gravity segregation</td>
<td>172</td>
</tr>
<tr>
<td>D.3 Shocks in analytical solution of 1D gravity segregation</td>
<td>173</td>
</tr>
<tr>
<td>D.4 Analytical solutions of gravity segregation with linear relative permeability curves</td>
<td>177</td>
</tr>
<tr>
<td>D.5 Evolution of solutions of gravity segregation with linear relative permeability curves</td>
<td>178</td>
</tr>
<tr>
<td>D.6 Final solutions of gravity segregation with linear relative permeability curves</td>
<td>179</td>
</tr>
<tr>
<td>D.7 Shocks to construct analytical solution for flux function with quadratic relative permeability curves</td>
<td>181</td>
</tr>
<tr>
<td>D.8 Analytical solutions of gravity segregation with quadratic relative permeability curves</td>
<td>183</td>
</tr>
<tr>
<td>D.9 Evolution of solutions of gravity segregation with quadratic relative permeability curves</td>
<td>185</td>
</tr>
<tr>
<td>D.10 Final solutions of gravity segregation with quadratic relative permeability curves</td>
<td>186</td>
</tr>
<tr>
<td>D.11 Flux functions of viscous and buoyancy forces</td>
<td>187</td>
</tr>
<tr>
<td>D.12 Initial condition of displacement under viscous and buoyancy forces</td>
<td>188</td>
</tr>
<tr>
<td>D.13 Evolution of solutions of displacement under viscous and buoyancy forces with linear relative permeability curves</td>
<td>189</td>
</tr>
<tr>
<td>D.14 Final solutions of displacement under viscous and buoyancy forces with linear relative permeability curves</td>
<td>190</td>
</tr>
</tbody>
</table>
D.15 Evolution of solutions of displacement under viscous and buoyancy forces with quadratic relative permeability curves .......................... 193
D.16 Final solutions of displacement under viscous and buoyancy forces with quadratic relative permeability curves .......................... 194
E.1 Solution for single-cell transport with viscous forces from continuation Newton ............................................................... 199
E.2 Solution for single-cell transport with viscous and buoyancy forces from CN ................................................................. 201
F.1 Flux functions for viscous fluxes .................................................. 204
F.2 Solutions of trust-region Newton method for 1D transports with viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). ......................... 207
F.3 Solutions of trust-region Newton method for 1D transport with viscous forces. \( k_{rw} = S^3 \) and \( k_{rn} = (1 - S)^3 \). ......................... 210
F.4 Solutions of trust-region Newton method for 1D transport with viscous forces. \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^2 \). ......................... 212
F.5 Solutions of trust-region Newton method for 1D transport with viscous forces. \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^{10} \) ......................... 214
F.6 Flux functions for viscous and buoyancy fluxes. \( N_g = -5 \) ........ 215
F.7 Solutions of trust-region Newton method for 1D transport with viscous and buoyancy forces. \( N_g = -5 \). \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). ........ 218
F.8 Solutions of trust-region Newton method for 1D transport with viscous and buoyancy forces. \( N_g = -5 \). \( k_{rw} = S^3 \) and \( k_{rn} = (1 - S)^3 \). ........ 220
F.9 Flux functions for gravity segregation ............................................. 221
F.10 Solutions of trust-region Newton method for 1D gravity segregation. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \) ............................................. 224
F.11 Solutions of trust-region Newton method for 1D gravity segregation.

\[ k_{rw} = S^3 \text{ and } k_{rn} = (1 - S)^3. \]

F.12 Permeability of SPE 10 top layer

F.13 Solution of saturation for transport in SPE 10 top layer, \( T = 10 \) days

F.14 Solution of saturation for transport in SPE 10 top layer, \( T = 100 \) days

F.15 Permeability of SPE 10 bottom layer

F.16 Solution of saturation for transport in SPE 10 bottom layer, \( T = 10 \) days

F.17 Solution of saturation for transport in SPE 10 bottom layer, \( T = 100 \) days
Chapter 1

Introduction

Numerical modeling of multiphase flow in subsurface porous media with application to oil-recovery and CO$_2$ sequestration processes is often referred to as Reservoir Simulation, and the computer program that is used for reservoir simulation is called a reservoir simulator. Reservoir simulation is now an essential tool for quantitative reservoir management in the oil and gas industry. It has been applied to improve petroleum production, design optimal development plans, understand reservoir features through history matching, and assess uncertainties associated with production decisions.

In order to simulate fluid flow in the reservoir, the mathematical model, including the governing equations, initial and boundary conditions, is constructed. The governing equations are discretized in time and space, and the resulting systems of nonlinear algebraic equations must be solved at every time step. The development of robust and efficient solvers for the nonlinear system of equations that describe flow and transport in large-scale, heterogeneous reservoir models is the objective of this dissertation.

The rest of this chapter is organized as follows. In Section 1.1, we state the mathematical model, i.e., The Partial Differential Equations (PDEs), that describe
nonlinear, two-phase immiscible, flow and transport in heterogeneous porous media. In Section 1.2, we discuss the motivations for our work and the challenges for nonlinear solvers. Then, we outline the remainder of the thesis and summarize our contributions in Section 1.3.

1.1 Mathematical model

We consider nonlinear immiscible, incompressible, two-phase flow in porous media. The conservation law for the two phases - referred to as nonwetting and wetting - can be written as:

\[ \phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, \]

where \( \phi \) is the porosity of the medium. We use subscript \( \alpha \) to denote the phases, i.e., \( w \) or \( n \). \( S_\alpha \) is the saturation, \( \mathbf{u}_\alpha \) is the velocity, and \( q_\alpha \) is the source term. The phase velocity is given by Darcy’s law:

\[ \mathbf{u}_\alpha = -k_{r\alpha} \frac{k_0}{\mu_\alpha} \left( \nabla p_\alpha + \rho_\alpha g \nabla h \right), \quad \alpha = w, n \]

where \( p_\alpha \) is the pressure, \( \rho_\alpha \) is the density, \( h \) is the height, \( k_{r\alpha} = k_{r\alpha}(S_\alpha) \) is the relative permeability, and \( \mu_\alpha \) is the viscosity.

We define the phase (relative) mobilities as \( \lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha} \). The end-point mobility ratio is defined as

\[ M^0 = \frac{k^0_{rw}/\mu_w}{k^0_{rn}/\mu_n}, \]

where \( k^0_{rw} = k_{rw}(1 - S_{nr}) \) and \( k^0_{rn} = k_{rn}(S_{wr}) \) are the maximum (end-point) relative permeabilities.

Substituting the phase velocities into Eqn. 1.1 and 1.2 we obtain two mass-balance
equations with four unknowns: $S_w$, $S_n$, $p_w$, and $p_n$. To close the system, we add the following relations:

\[
S_w + S_n = 1 \quad \text{(saturation constraint),} \\
p_n - p_w = p_c \quad \text{(capillary-pressure relation),}
\]

where the capillary pressure $p_c = p_c(S_w)$ is a nonlinear function of the wetting saturation.

Substitution of Eqn. 1.3 into Eqns. 1.1 and 1.2 yields a coupled system of nonlinear parabolic equations. The system often exhibits a mixed near elliptic-hyperbolic character, which becomes apparent when we sum up Eqns. 1.1 and 1.2 to obtain the pressure equation. We can use the saturation constraint in the summation of Eqns. 1.1 and 1.2 to get

\[
\nabla \cdot \mathbf{u}_t = q_t, \tag{1.7}
\]

where

\[
\mathbf{u}_t = -k\lambda_t \nabla p_w - kg (\lambda_w \rho_w + \lambda_n \rho_n) \nabla h - k\lambda_n \nabla p_c, \tag{1.8}
\]

and

\[
q_t = q_w + q_n. \tag{1.9}
\]

Here, $\lambda_t = \lambda_w + \lambda_n$ is the total mobility. Substitution of Eqn. 1.8 into Eqn. 1.7 gives the pressure equation, which is an elliptic PDE. In the absence of source/sink terms, Eqn. 1.7 indicates that the total-velocity $\mathbf{u}_t$ is divergence-free, i.e.,

\[
\nabla \cdot \mathbf{u}_t = 0. \tag{1.10}
\]

We can rewrite the wetting-phase velocity in terms of the total-velocity as

\[
\mathbf{u}_w = \frac{\lambda_w}{\lambda_t} \mathbf{u}_t - k g \frac{\lambda_w \lambda_n}{\lambda_t} (\rho_w - \rho_n) \nabla h + k \frac{\lambda_w \lambda_n}{\lambda_t} \nabla p_c. \tag{1.11}
\]
In Eqn. 1.11, if the total-velocity is constant, $u_w = u_w(S_w)$ is then a function of saturation only. Substituting Eqn. 1.11 into Eqn. 1.1, the transport (saturation) equation is obtained as:

$$\varphi \frac{\partial S_w}{\partial t} + \nabla \cdot \left( \frac{\lambda_w}{\lambda_t} u_t - k g \frac{\lambda_w \lambda_n}{\lambda_t} (\rho_w - \rho_n) \nabla h + k \frac{\lambda_w \lambda_n}{\lambda_t} \nabla p_c \right) = 0; \quad (1.12)$$

Subject to proper initial and boundary conditions, the saturation distribution can be obtained by solving this PDE. Combining the pressure equation (Eqn. 1.7) with the transport equation (Eqn. 1.12), we obtain the flow-transport system for immiscible, incompressible, two-phase flow.

For flow in one dimension (1D) and assuming that the total velocity, $u_t$, is constant, the transport equation can be written as

$$\varphi \frac{\partial S_w}{\partial t} + u_t \frac{\partial F_w}{\partial x} = 0. \quad (1.13)$$

Here, $F_w = \frac{u_w}{u_t}$ is the flux (fractional flow) of the wetting phase. It is defined as

$$F_w = \frac{\lambda_w}{\lambda_t} - k g \frac{\lambda_w \lambda_n}{\lambda_t} (\rho_w - \rho_n) \frac{\nabla h}{u_t} + k \frac{\lambda_w \lambda_n}{\lambda_t} \frac{\nabla p_c}{u_t}, \quad (1.14)$$

We introduce two dimensionless quantities:

$$N_g = \frac{k(\rho_w - \rho_n)g \nabla h}{\mu_n u_t}, \quad (1.15)$$

and

$$P_e = \frac{u_t \mu_n L}{k \bar{\rho}_c}, \quad (1.16)$$

where $N_g$ is the gravity number, which is the ratio of buoyancy to viscous forces. Here we assume that the $z$ coordinate is pointing upward, and use $h$ to denote the height. Therefore we have $\nabla h > 0$. We also assume that the wetting phase is heavier, i.e.,
\[ \rho_w > \rho_n. \]  
\( P_e \) is the Peclet number, which is the ratio of viscous to capillary forces.  
\( L \) is a characteristic length scale, and \( \bar{p}_c \) is a characteristic capillary pressure. With these definitions, \( F_w \) can be written as

\[
F_w = \frac{\lambda_w}{\lambda_t} - \frac{k_{rn} \lambda_w}{\lambda_t} N_g + \frac{\lambda_w k_{rn} \nabla p_c}{\bar{p}_c/L} \frac{1}{P_c}. \tag{1.17}
\]

The three terms on the right-hand side account for the viscous, buoyancy, and capillary fluxes, respectively. We denote the flux that accounts for both the viscous and buoyancy forces as

\[
f_w = \frac{\lambda_w}{\lambda_t} - \frac{k_{rn} \lambda_w}{\lambda_t} N_g. \tag{1.18}
\]

Since the total-velocity is assumed constant, we can make the transport equation dimensionless by defining

\[
t_D = \frac{tu_t}{\phi L} \tag{1.19}
\]

and

\[
x_D = \frac{x}{L}. \tag{1.20}
\]

Then, the 1D dimensionless transport equation is

\[
\frac{\partial S_w}{\partial t_D} + \frac{\partial F_w}{\partial x_D} = 0. \tag{1.21}
\]

From here on, we drop the subscripts, so we can write the equation in a more concise form:

\[
\frac{\partial S}{\partial t} + \frac{\partial F}{\partial x} = 0. \tag{1.22}
\]

In the absence of gravity and capillarity, \( F = \frac{\lambda_w}{\lambda_t} \) and is a monotonic function of saturation. For multiphase flow in porous media, the flux function, \( F = F(S) \) in Eqn. 1.22 is usually S-shaped \[1,2\] (i.e., not uniformly convex or concave). An S-shaped flux function is shown in Fig. 1.1(a). Inflection points are represented by
red dots in Fig. 1.1(a) The presence of gravity, or capillarity, changes the shape of the flux function. When buoyancy is dominant, the flux function becomes non-monotonic indicating the occurrence of counter-current flow for part of the saturation range. Note that the convexity of the flux function changes as \(N_g\) and \(P_e\) change. Fig. 1.1(b) shows a typical example of the flux function with strong buoyancy, e.g., \(N_g = -5\). In the figure, there are two inflection points (red dots in Fig. 1.1(b)), a sonic point (green dot), and a unit-flux point (orange dot). The sonic point is where the flux function is maximum; the unit-flux point is where the flux is unity at a point \(S < 1\). Fig. 1.1(c) shows that the presence of strong capillary forces (e.g., \(P_e = 0.2\)) changes the shape, including the inflection point, of the flux function. This is due to the nonlinear diffusion term [3], i.e., \(\nabla \rho_c(S_w)\) in the third term of \(F_w\) (Eqn. 1.14), introduced into the phase flux by capillarity. This diffusion term tends to mix the two fluid phases together if there is a saturation gradient.

Figure 1.1: flux functions: (a) Viscous flux \((M^0 = 1, N_g = 0, P_e = \infty)\); (b) Viscous and buoyancy flux \((M^0 = 1, N_g = -5, P_e = \infty)\); (c) Viscous and capillary flux \((M^0 = 1, N_g = 0, P_e = 0.2)\). Red dots represent inflection points; orange dots are unit-flux points; and green dots are sonic points.
1.2 Challenges for nonlinear solvers

In reservoir simulation, the use of explicit time integration schemes often leads to severe restrictions on the timestep size \[4, 5, 1, 6\]. When a large-scale heterogeneous reservoir model is simulated, it is often the case that for a given global timestep size, the Courant-Friedrichs-Lewy (CFL) numbers in the computational domain can vary by orders of magnitude \[7, 8\]. In such cases, the use of explicit time integration schemes is simply not feasible, and implicit time integration is required. The backward-Euler implicit scheme yields a system of nonlinear discrete conservation equations. The nonlinear system is usually cast in residual form and solved using the Newton method \[1\]. The Newton method involves a sequence of iterations, each involving the construction of the Jacobian matrix and solution of the resulting linear system:

\[
J(S^\nu)\delta S^{\nu+1} = -R(S^\nu)
\]  

(1.23)

where \(S^\nu\) is the unknown vector (e.g., saturation) at the current iteration, \(R(S^\nu)\) is the residual function, and \(J(S^\nu) = (\partial R/\partial S)|_{S^\nu}\) is the Jacobian matrix. By solving this linear system, we may obtain the update vector \(\delta S^{\nu+1}\), which is used to update the unknown vector as:

\[
S^{\nu+1} = S^\nu + \delta S^{\nu+1}
\]

(1.24)

where \(S^{\nu+1}\) is the unknown vector at the next iteration. This process is performed until the solution of the nonlinear system is obtained for the target timestep. The Newton method is popular because of its general applicability and local quadratic convergence \[1, 5\]. For the residual equations arising from discretized PDEs, the resulting Jacobian is generally sparse, which means the linear systems can be solved efficiently. Also, quadratic convergence means that the Newton method converges rapidly when good initial guesses are available.
The backward-Euler scheme, which is referred to as the Fully Implicit Method (FIM), is unconditionally stable, but there is no guarantee that the Newton solver will converge \[1, 6\]. In reservoir simulation, heuristic techniques to control the timestep size are used \[9, 10\]. The use of such heuristics can lead to timestep sizes that are too conservative resulting in unacceptably large computational time and wasted computations; more importantly, the heuristics are based on tuning with trial-and-error and are not guaranteed to work \[11\].

The objective of this work is to develop robust and efficient nonlinear solution algorithms for immiscible two-phase flow and transport in the presence of viscous, buoyancy, and capillary forces. Fig. 1.2 is a simple depiction of the parameter space. The vertices of the triangle represent the viscous, gravitational, and capillary forces. The edges represent the combination of two mechanisms, and the interior involves all three mechanisms. Our focus here is on modeling flow processes in heterogeneous domains, where all three mechanisms play important roles in the evolution of the pressure and saturation field as a function of space and time.

An important step toward the development of a Newton-based solver that is especially tuned for nonlinear transport in porous media was taken by Jenny et al. \[12\]. They provided convincing evidence that the nonlinearity of the mass conservation laws is dominated by the nonlinearity of the flux function, which can be localized and resolved efficiently. Jenny et al. \[12\] proposed a modified Newton solver, which was proved to be unconditionally convergent for single-cell transport problems in the presence of viscous forces only.

Another important step toward more rigorous nonlinear solvers is the reduced Newton method with potential-based ordering strategy proposed by Kwok and Tchelepi \[8\]. By using the potential-based ordering, then instead of solving the coupled large-scale flow and transport problem (potentially with millions degrees of freedom), we
CHAPTER 1. INTRODUCTION

solve a series of single-cell transport problems. Specifically, we solve for the saturations of the cells from the highest phase potential to the lowest phase potential, one cell at a time. The potential-based ordering strategy and the reduced-Newton method will be discussed in more details in Section 2.2.

With the help of the ordering strategy in [8], the modified Newton method in [12] can be applied for general-purpose large-scale reservoir simulation. Voskov and Tchelepi [13] used numerical examples to demonstrate that for immiscible gas displacement problems in large-scale heterogeneous reservoir models, the modified Newton method can increase the maximum convergent timestep size by at least one order of magnitude compared with the standard Newton method. Therefore, there is compelling evidence that resolving the nonlinearity of the flux function is the key for efficient large-scale simulation.

In this dissertation, we take a significant step further. We propose a Newton solver based on resolving the nonlinearity for the immiscible two-phase transport
problem across the entire viscous-buoyancy-capillary parameter space. To deal with the nonlinearity for the flux function in the presence of strong buoyancy and/or capillary forces, we step back and investigate single-cell and 1D transport problems.

In 1D, we can usually assume the total-velocity to be a constant. Thus, the saturation distribution can be obtained by solving the transport problem (saturation equations) only. The saturation equations for multiphase transport are highly nonlinear and are characterized by non-convex flux functions that can be non-monotonic in the presence of strong buoyancy. Once the convergence difficulties in the nonlinear transport problem get resolved, the simulation in 1D can be performed with timestep sizes that are only based on accuracy considerations.

However, in 2D or 3D domains, the assumption of constant total-velocity is no longer valid. We need to solve the coupled flow (pressures and velocities) and transport (saturation) system for each timestep. The coupling is due to the strong dependence of the saturation field on the fractional flow and total velocity, which is calculated from the pressure distribution. As discussed in Section 1.1, the coupled system usually exhibits a mixed near elliptic-hyperbolic character [1, 14]. Many numerical formulations are designed by taking this mixed character into consideration, where the coupled system can be separated into elliptic (pressure) and near-hyperbolic (saturation) parts [15, 5, 6].

In the last few years, there has been significant progress in our ability to solve the nearly-elliptic pressure equation associated with multiphase flow in heterogeneous formations. Examples include algebraic multi-grid (AMG) [16, 17], multiscale finite-element methods (MsFEM) [18, 19], and the multiscale finite-volume (MSFV) method [20, 21, 22]. More recently, stable and convergent schemes have been developed for nonlinear transport in one dimension [23, 8, 14]. Unconditionally convergent schemes have been developed for the nonlinear saturation equation for viscous dominated flows [12]. A major ongoing challenge is how to resolve the coupling between flow
and transport efficiently when viscous and buoyancy forces are both present and significant. The two-stage CPR (Constrained Pressure Residual)\cite{24, 25} preconditioning strategy is an example of resolving the coupling at the linear level. On the nonlinear level, the sequential-implicit method deals with flow and transport separately and differently, which is suitable for and hence employed in multiscale methods, such as MSFV\cite{22, 26}.

In the sequential-implicit method (SIM), the overall problem is split into two parts: flow and transport. For the flow problem, the mass conservation equations are combined and written in terms of pressure (Eqn. 1.7), which is then solved implicitly. Then, the total-velocity is calculated using the pressure field. The total-velocity is then fixed in the transport problem (Eqn. 1.12), which is solved implicitly. For each timestep, there are two inner loops: one for pressure and one for saturation, and there is an outer loop. For each iteration of the outer loop, the computations proceed as follows: compute the pressure field iteratively to a certain tolerance, update the total-velocity, then compute the saturation iteratively. Extensive numerical experience indicates that SIM works very well for viscous-dominated problems. However, SIM suffers from serious difficulties in the presence of buoyancy and strong capillarity. When buoyancy forces are significant, the challenge lies in the fact the flow directions of the two phases (at the interface between two cells) based on the pressure solution may change once the saturation is updated by solving the transport problem. Changes of the phase-flow direction between the pressure and saturation solutions (or updates) can slow convergence of the outer-loop quite significantly. In cases where aggressively large timesteps (CFL $\gg 1$) are taken in the presence of significant buoyancy, SIM may not converge at all, thus requiring timestep cuts. It is observed numerically in \cite{27} that when the MSFV formulation is embedded into the sequential-implicit method to model multiphase flow in heterogeneous porous media with strong gravity, extremely small timesteps (CFL $\ll 1$) have to be used. In order
CHAPTER 1. INTRODUCTION

To resolve this issue and enhance the convergence and stability of SIM for coupled flow and transport, we propose a nonlinear solution strategy that deals effectively with convergence difficulties associated with flow reversal between Newton iterations during a timestep.

1.3 Thesis outline

This dissertation is organized as follows. In Chapter 2, we review the state-of-the-art for nonlinear solvers of multiphase flow in porous media, including modified Newton ([12] and [13]), reduced-Newton ([8]) and continuation-Newton methods ([28]). These nonlinear solvers tackle the nonlinear problem from different perspectives, and they serve as the starting point for this dissertation. In Chapter 3, we present a new nonlinear solver for transport in heterogeneous porous media. The solver employs trust regions of the flux function to guide the Newton iterations and solution updating. The new nonlinear solver is unconditionally convergent for single-cell transport and overcomes the often severe limits on timestep size associated with existing Newton-based solvers. In Chapter 4, we analyze the convergence difficulties associated with coupled flow and transport due to flow reversal, and we propose a strategy to resolve this issue. In Chapter 5, we propose a preconditioning strategy to overcome convergence difficulties in the nonlinear solver that are associated with the propagation of saturation fronts into regions at, or near, residual saturations. Finally, we present our conclusions and outline future directions in Chapter 6.
Chapter 2

Nonlinear Solvers: State of the Art

The fully implicit method (FIM) for the discretization of the governing equations (Eqn. 1.1 and 1.2) leads to nonlinear systems that must be solved at each timestep. For general problems, Newton’s method is not guaranteed to converge, and it is known to be sensitive to the initial guess [29, 30, 1]. In most reservoir simulators, the initial guess is the old state (i.e., the pressure and saturation distributions from the previous timestep). For small timestep sizes, this is a good approximation of the new state and, therefore, is likely to be a good starting point for the Newton iteration. For larger timesteps, however, the old state may not be a good initial guess, and the iterations may converge too slowly, or even diverge.

To overcome the convergence difficulties of Newton’s method, empirical techniques for timestep control are utilized in reservoir simulators ([1] [17]). With a try-adapt-try strategy, an attempt to solve for a chosen timestep is made. If that fails within a specified number of (Newton) iterations, the timestep is adapted heuristically, and the process is restarted; thus, the previous effort is wasted.

In this chapter, we study the nature of the nonlinearities that are typical of porous media flows, and we review the state-of-the-art in reservoir simulation practice.
2.1 Modified Newton method

Local Newton methods refer to the situation that ‘sufficiently good’ initial guesses of the solutions are assumed to be available \[30\]. That is, local Newton methods are expected to converge only when the initial guess is close enough to the solution. On the other hand, global Newton methods refer to Newton-like methods that are convergent independently of the initial guess. The multiphase porous-media transport problem has strong nonlinearity, and the standard Newton method is convergent only when the timestep size is ‘sufficiently small’ and the initial guess is close to the solution. A globalization technique \[30\] is needed to improve the convergence behavior of Newton-based methods, when the initial guess is not close to the solution.

Jenny et al. \[12\] proposed a modified Newton method for hyperbolic conservation laws in the absence of gravity and capillarity. They proposed a simple chopping scheme within the Newton loop results in a nonlinear solver that is convergent for arbitrarily large timestep sizes, and hence allowing one to choose the timestep size based on accuracy considerations only. A brief description of the Jenny et al. \[12\] method follows.

The degree of nonlinearity of the residual for the transport problem, Eqn. \[1.22\], is strongly related to the shape of the flux function, especially when the timestep is large \[31\]. For viscous-dominated multiphase flow, the nonlinear flux function is usually S-shaped (Fig. \[1.1(a)\]). The inflection point is the saturation where the flux function has the largest slope. As expected, if the solution (saturation) resides on one side of the inflection point, whereas the initial guess is on the other side, it can be hard for the Newton iterative process to converge. Jenny et al. \[12\] proposed the following: if the Newton update would cross the inflection point, it is scaled back to the inflection point, and the Newton process is continued based on the scaled back (chopped) update. Note that the Newton update is not scaled back exactly to the
inflection point. Instead, it is scaled back to one side of the inflection point, i.e., $S_{\text{inflec}} \pm \epsilon$, to make sure that the two successive updates reside on the same side of the inflection point. The flow chart of this modified Newton method is presented in Fig. 2.1.

![Flow chart of the modified Newton method of Jenny et al.](image)

Figure 2.1: Flow chart of the modified Newton method of Jenny et al. [12], for one timestep. Solution of the linearized transport equation is represented by the operator $T$.

This flux-based Newton method has proved to be quite powerful for the simulation of immiscible viscous-dominated displacements in large-scale heterogeneous models. Its applicability and efficiency for general-purpose compositional simulation was demonstrated by Voskov and Tchelepi in [13], who extended the approach to solve compositional problems that employ the molar (or mass) variables. They showed that the flux functions associated with the key tie-lines play a dominant role in the evolution of the solution. For a four-component gas injection problem in the top eight layers of the SPE 10 model without gravity, the modified flux-based Newton scheme is shown to be always more stable and converges faster than the safeguarded Newton method, which employs heuristics on maximum changes in the variables ([32]). The gas saturation at the end of the simulation for immiscible gas displacement in shown in Fig. 2.2.
To handle multiphase flow with significant buoyancy and/or capillary forces, we need to analyze the influence of the nonlinearity of the flux function in the presence of buoyancy and/or capillarity on the nonlinear solver. The shape and features of the flux function were described at the end of Section 1.1. The flux-based nonlinear solver that handles buoyancy and capillarity is discussed in the next chapter.

### 2.2 Reduced-Newton method

Kwok and Tchelepi [8] proposed a potential-based ordering of the equations and unknowns that allows one to solve for the saturations one cell at a time. The proposed ordering is valid for both two-phase and three-phase flow and for viscous, buoyancy, and capillary forces. For a two-phase system where the transport equations are discretized by a standard, implicit, upstream mobility-weighted scheme (standard FIM),
the nonlinear system can be arranged in the following form

\[
\begin{align*}
    f_{w1}(S_1, & \quad p_1, \cdots, p_N) = 0 \\
    f_{w2}(S_1, S_2, & \quad p_1, \cdots, p_N) = 0 \\
    \vdots \quad & \\
    f_{wN}(S_1, S_2, \cdots, S_N, & \quad p_1, \cdots, p_N) = 0 \\
    f_{o1}(S_1, & \quad p_1, \cdots, p_N) = 0 \\
    f_{o2}(S_1, S_2, & \quad p_1, \cdots, p_N) = 0 \\
    \vdots \quad & \\
    f_{oN}(S_1, S_2, \cdots, S_N, & \quad p_1, \cdots, p_N) = 0
\end{align*}
\]

where \( p_i \geq p_j \) whenever \( i < j \). The monotonicity of the pressure field guarantees that the transport equations for cell \( j \) depend only on the saturations \( S_i \) with \( i \leq j \). The triangular structure carries over to the Jacobian, which now has the form

\[
J = \begin{bmatrix}
S_w & p \\
J_{ww} & J_{wp} \\
J_{ow} & J_{op}
\end{bmatrix}
\]  

where \( J_{ww} \) is lower triangular.

Based on the above potential-based ordering, a reduced-Newton method is proposed in [8]. Within each iteration, pressure is first updated by solving the following reduced Jacobian:

\[
J_{\text{reduced}} = J_{op} - J_{ow}J_{ww}^{-1}J_{wp}
\]  

(2.2)

Since \( J_{ww} \) is lower triangular, the reduced Jacobian can be constructed efficiently. Note that the pressure solution obtained here is identical to the one obtained from the fully-implicit method. Then, for the updated pressure field, the saturations are
updated cell by cell. That is, we solve for the saturations of the cells with the highest potential (e.g., the cells perforated by injectors) first, and then proceed to solve for saturations at the downstream of these cells according to the phase potential. This process continues until the saturations at the cells with the lowest potential (e.g., the cells perforated by producers) have been updated. The reduced-Newton algorithm is summarized in Fig. 2.3.

1. While $|F_o(S(p^k), p^k)| > tol$, do
2. Form the full Jacobian $J = \begin{bmatrix} J_{ww} & J_{wp} \\ J_{ow} & J_{op} \end{bmatrix}$ evaluated at $(S(p^k), p^k)$;
3. Solve $(J_{op} - J_{ow} J_{ww}^{-1} J_{wp}) \delta p^k = -r^k$;
4. Compute $p^{k+1} = p^k + \delta p^k$;
5. Update $S^{k+1} = S(p^{k+1})$ nonlinearly by solving $F_w(S^{k+1}, p^{k+1}) = 0$, one variable at a time in potential order;
6. $k := k + 1$
7. end

Figure 2.3: Flow chart for the reduced-Newton method (from Kwok and Tchelepi [8])

Numerical evidence in [8] shows that the potential-based reduced-Newton solver is able to converge for time steps that are much larger than what the standard Newton method can handle. In addition, when both methods can converge, the nonlinear solver in [8] converges faster than the standard Newton strategy.

The phase-based potential ordering strategy in [8] provides us with the opportunity to resolve the nonlinearity for single-cell problems first, and then extend the methodology derived for single-cell problems to large-scale simulation. Therefore, to obtain a nonlinear solution strategy that is convergent for large-scale transport problems, we can start by analyzing the nonlinearity of single-cell transport problems. Such an analysis is described in the next chapter.
2.3 Continuation-Newton method

Continuation (homotopy, or embedding) methods ([33, 34]) are nonlinear solvers that associate a timestep with each iteration. These approaches converge when the residual drops below a certain threshold and the associated timestep reaches the target timestep. Continuation methods solve the nonlinear equations \( R(u, \lambda) = 0 \) for various values of a real parameter \( \lambda \). For numerical continuation, a solution path is traced out using a predictor-corrector, path-following technique. The parameter \( \lambda \) is repeatedly incremented until the desired value is reached. In each iteration, the current solution \( u \) is used as an initial iterate. For a detailed review of continuation methods, see [33, 34, 35, 36].

Younis et al. ([11]) developed a Continuation-Newton (CN) method that solves the implicit residual system using a combination of the Newton method and continuation on the timestep size. In [11] and [28], a continuation-based solution process that associates a timestep size with each iteration is formulated, i.e., the timestep size is a parameter which is continuously changing. The CN method of Younis et al. follows the solution path loosely. A more detailed description of CN follows.

In the nonlinear problem

\[
R \left( S^{n+1}, \Delta t; S^n \right) = 0, \tag{2.3}
\]

\( R \) is the vector of discrete residual equations and \( S \) is the saturation. The solution path can be written as:

\[
S^{n+1} = S^{n+1} (\Delta t), \tag{2.4}
\]

which is continuous and emanating from the condition at the previous timestep \( S^n \).
For the solution path, we have

$$\frac{dS^\nu}{d\Delta t} = -J(S^\nu, \Delta t; S^n)^{-1} \frac{\partial R(S^\nu, \Delta t; S^n)}{\partial \Delta t},$$

(2.5)

i.e., the tangent of the solution path is known.

An illustration of solution path with only one unknown is shown in Fig. 2.4. In Fig. 2.4, the solution path emanates from the initial condition \((S = S^n, \Delta t = 0)\), and continues to the target time step, \(\Delta t_{\text{target}}\), augmented with its solution, \(S^{n+1}\).

Figure 2.4: Illustration of solution path and iterative solutions for Newton method (from Younis [28]). Note the iterative solutions are evaluated at the target timestep.

The proposed algorithm in [11] and [28] defines a convergence neighborhood around the solution path (illustrated in Fig. 2.5). Any point inside the neighborhood is considered to be a good estimate of the solution. In Fig. 2.5, it is shown that for each iteration in CN, the solution is obtained either by a tangent prediction (e.g., from \(p_0\) to \(p_1\), or from \(p_1\) to \(p_2\)) or by a (Newton) correction (e.g., from \(p_2\) to \(p_3\)).
CHAPTER 2. NONLINEAR SOLVERS: STATE OF THE ART

For each tangent step, the step length is chosen, such that the next solution remains within the convergence neighborhood. A correction step is triggered in order to bring the solution closer to the solution path, if the next tangent step-length would be too small, or zero. The algorithm guarantees convergence for any timestep size. If the iteration process is stopped before the target timestep is reached, the last iterate is a solution to a smaller and known timestep.

![Illustration of iterative solutions using the continuation-Newton approach](image)

Figure 2.5: Illustration of iterative solutions using the continuation-Newton approach (from Younis [28]). In the illustration, two tangent steps are followed by a correction step.

2.4 Ordinary trust-region method: NLEQ-RES algorithm

The standard Newton method can be algebraically derived by linearization of the nonlinear equation around the solution point \( S^{n+1} \). This kind of derivation supports
the interpretation that the Newton correction is useful only in a close neighborhood of \( S^{n+1} \). Far away from \( S^{n+1} \), such a linearization might still be trusted in some ‘trust region’ around the current iterate \( S^\nu \). There are several models defining such a region. One of them is Levenberg-Marquardt model \([37, 38, 39]\). Empirical trust-region strategies for the Levenberg-Marquardt method have been worked out, e.g., by M. D. Hebden \([40]\), by J. J. Moré \([39]\), or by J. E. Dennis et al. \([41]\). The associated codes are rather popular and included in several mathematical software libraries. An affine contravariant reformulation of the Levenberg-Marquardt model leads us to study the damped Newton iteration:

\[
J(S^\nu)\Delta S^\nu = -R(S^\nu), \hspace{1cm} (2.6)
\]

\[
S^{\nu+1} = S^\nu + \lambda^\nu \Delta S^\nu, \hspace{0.5cm} \lambda^\nu \in [0, 1], \hspace{1cm} (2.7)
\]

under the requirement of residual contraction

\[
\|R(S^{\nu+1})\| < \|R(S^\nu)\|, \hspace{1cm} (2.8)
\]

which is certainly the most popular and the most widely used global convergence measure. There are theoretical analyses, which are characterized by means of affine contravariant Lipschitz conditions, to derive the theoretically optimal iterative damping factors and prove global convergence within some range around these optimal factors \([42]\). However, in the theoretical analyses, there are parameters needed for the calculation of damping factors, e.g., Lipschitz constant, that are computationally unavailable. Thus, theoretically optimal damping factors generally cannot be obtained in numerical computation. To get an algorithmic estimation of the damping factors, residual based trust-region strategies has been developed based on the theoretical analyses and computational estimates. Example codes are described in
Here, we focus on the global Newton method with residual based convergence criterion and adaptive trust-region strategy, so-called NLEQ-RES algorithm.

Note that the trust-region strategies in the NLEQ-RES are defined based on the local descent of residual, i.e., \( \| R(S^{\nu+1}) \| < \| R(S^\nu) \| \), without considering the global structure and nonlinearity of the residual function in any specific nonlinear problem. In the NLEQ-RES, the trust regions for current iteration, \( \nu+1 \), is derived based on the residual norm in the previous iteration, \( \| R(S^\nu) \| \), and the current iteration. Hence the definition of trust regions is history-based. Also, since \( R(S^{\nu+1}) \) is unavailable a priori, an attempt of try-adapt-try is necessary to obtain the solution for current iteration.

Compared to the NLEQ-RES and other algorithmic trust-region strategies based on Levenberg-Marquardt model in [30], our trust-region Newton method in Chapter 3 is based on the structure and nonlinearity of residual function (i.e., the flux function), specially, for immiscible two-phase flow and transport in porous media, in which the trust regions are delineated before solving the nonlinear problem.

The algorithm NLEQ-RES is described as follows:

**Algorithm 1 NLEQ-RES** [30]

**Require:** Guess an initial iterate \( S^0 \). Evaluate \( R(S^0) \). Set an initial damping factor \( \lambda_0 := \lambda_{\text{min}} \).

**for** iteration index \( \nu = 0, 1, \ldots \) **do**

Convergence test:

- **if** \( \| R(S^\nu) \| \leq \varepsilon \) **then**
  - Stop. Solution found \( S^{\nu+1} := S^\nu \).
- **else** Evaluate Jacobian matrix \( J(S^\nu) \). Solve linear system \( J(S^\nu)\Delta S^\nu = -R(S^\nu) \).

**end if**

**if** \( \nu > 0 \) **then** Compute a prediction value for the damping factor

\[
\lambda_{\nu} := \min(1, \mu_{\nu}),
\]

where \( \mu_{\nu} = \frac{\| R(S^{\nu-1}) \|}{\| R(S^\nu) \|} \mu'_{\nu-1} \).

**end if**
Regular test:
if $\lambda_\nu < \lambda_{\text{min}}$ then
    Stop. Convergence failure.
else
    Compute the trial iterate $S^{\nu+1} := S^\nu + \lambda_\nu \Delta S^\nu$ and evaluate $R(S^{\nu+1})$.
end if

Compute the monitoring quantities

$$\Theta_\nu := \frac{\|R(S^{\nu+1})\|}{\|R(S^\nu)\|},$$

$$\mu'_\nu := \frac{\|R(S^\nu)\|}{2\|R(S^{\nu+1})-(1-\lambda_\nu)R(S^\nu)\|}.$$ 

if $\Theta_\nu \geq 1$ then
    Replace $\lambda_\nu$ by $\lambda'_\nu := \min(\mu'_\nu, \frac{1}{2}\lambda_\nu)$. Go to Regularity test.
else: Let $\lambda'_\nu := \min(1, \mu'_\nu).
if $\lambda'_\nu \geq 4\lambda_\nu$ then
    Replace $\lambda_\nu$ by $\lambda'_\nu$ and goto 2.
else
    Accept $S^{\nu+1}$ as new iterate.
end if
$\nu \leftarrow \nu + 1$
end if
end for
Chapter 3

Trust-Region Newton Method for Two-phase Transport

The backward-Euler scheme is unconditionally stable, but there is no guarantee that the Newton solver will converge [11, 6]. In reservoir simulation, heuristic techniques to control the timestep size are used [9, 10]. The use of such heuristics often leads to timestep sizes that are too conservative resulting in unacceptably large computational time and wasted computations [11]. Our objective is to develop a nonlinear solver for multiphase transport in heterogeneous porous media that is unconditionally convergent, so that the timestep size is chosen solely on accuracy considerations without worrying about the robustness of the solver itself.

In this chapter, we present a new nonlinear solver for immiscible two-phase transport in porous media, where viscous, buoyancy, and capillary forces are significant. Based on detailed analysis of the nonlinearity, we propose a nonlinear transport solver that employs trust regions of the flux function to guide the Newton iterations and solution updating. We start from the finite-volume discretization of the nonlinear transport equation.
3.1 Discretized transport problem

In 1D (Fig. 3.1), the transport problem, Eqn. 1.22 can be discretized using a local conservative, low-order, finite-volume scheme, which can be written for control-volume (cell) $i$ as follows:

$$S_i^{n+1} - S_i^n + \frac{\Delta t}{\Delta x}(F_{i+\frac{1}{2}}^{n+1} - F_{i-\frac{1}{2}}^{n+1}) = 0. \quad (3.1)$$

Assuming the total-velocity is constant, and its direction is from left to right, the discretized flux evaluated at interface $i + \frac{1}{2}$ is

$$F_{i+\frac{1}{2}}^{n+1} = f(S_i^{n+1}; S_{i+1}^{n+1}) + \frac{C(S_{i+1}^{n+1}) - C(S_i^{n+1})}{\Delta x}, \quad (3.2)$$

where $f$ is the viscous and gravitational flux at the cell interface, and $C$ is a function that accounts for capillarity. The capillary flux is evaluated as a central difference of $C(S)$ [3], where capillarity is treated as a diffusion term (i.e., capillary mixing due to the saturation difference) in the conservation law.

For the viscous and gravitational flux, Kwok and Tchelepi [44] proved that the fully implicit scheme with phase-based upstream weighting converges to the entropy solution of the conservation law irrespective of the CFL number. So, in Eqn. 3.2, the viscous and gravitational fluxes at cell interfaces are evaluated using phase-based
upstream weighing. Namely, \[45\]

\[ f_{i+\frac{1}{2}}(S_i^{n+1}, S_{i+1}^{n+1}) = \begin{cases} 
\frac{Mk_{rw}(S_i)[1-N_gk_{rn}(S_i)]}{Mk_{rw}(S_i)+k_{rn}(S_i)}, & 0 \leq S_i \leq S_f = 1 \\
\frac{Mk_{rw}(S_i)[1-N_gk_{rn}(S_{i+1})]}{Mk_{rw}(S_i)+k_{rn}(S_{i+1})}, & S_f = 1 < S_i \leq 1 
\end{cases} \tag{3.3} \]

where \( M = \frac{\mu_n}{\mu_w} \) is the viscosity ratio. Here, \( S_f = 1 \) is the saturation point that corresponds to \( f = 1 \). Obviously, in the absence of counter-current flow, \( S_f = 1 = 1 \). Eqn. 3.3 tells us that when \( S_i \) is smaller than \( S_f = 1 \), both the wetting and nonwetting phases move from cell \( i \) to cell \( i + 1 \). Hence, the interface flux, \( f_{i+\frac{1}{2}} \), is evaluated with the phase mobilities (or relative permeabilities) of cell \( i \). On the other hand, when \( S_i \) is larger than \( S_f = 1 \), the wetting move moves from cell \( i \) to cell \( i + 1 \), but the nonwetting phase moves from cell \( i + 1 \) to cell \( i \). So, \( f_{i+\frac{1}{2}} \) is evaluated with \( k_{rw}(S_i) \) and \( k_{rn}(S_{i+1}) \).

For the capillary flux, an integral form was introduced by Douglas et al. in \[46\] and also used by Cances in \[3\]. Specifically,

\[ C(S) = \frac{1}{P_c/L} \frac{\mu_n}{P_c} \int_0^S \frac{\lambda_w(u)\lambda_n(u)}{\lambda_w(u) + \lambda_n(u)} p_c'(u) du. \tag{3.4} \]

When \( S \) approaches zero, both \( p_c(S) \) and \( p_c'(S) \) (i.e., \( \frac{dp_c}{dS} \)) are close to infinity, and that leads to serious numerical difficulties. In contrast, \( C(S) \) is well-behaved, i.e., bounded, when \( S \) goes to zero. Note that when \( S \) approaches zero, \( p_c' \) approaches infinity and \( \frac{\lambda_w\lambda_n}{\lambda_w + \lambda_n} \) must approach zero such that the asymptotic limit of \( \frac{\lambda_w\lambda_n}{\lambda_w + \lambda_n} p_c' \) is bounded. That is, \( \frac{\lambda_w\lambda_n}{\lambda_w + \lambda_n} \) must approach zero asymptotically faster than \( p_c' \) approaching infinity. This hypothesis is quite reasonable and is satisfied when the relative permeability is a quadratic function of saturation, and the Brooks-Corey model (\[47\]) is used for the capillary pressure. See appendix A.1 for a detailed analysis.

It is shown in \[3\] that Eqn. 3.2 is applicable if the capillary pressure function is the same for the cells (control volumes) on either side of the interface. If the two cells
have different capillary-pressure curves (heterogeneous capillarity), we need to solve for a pair of ‘dummy’ saturations on either side of the interface, \( S^{n+1}_{i+\frac{1}{2},L} \) and \( S^{n+1}_{i+\frac{1}{2},R} \) to ensure continuity of both the (wetting) phase flux and the capillary pressure at the interface. That is,

\[
F^{n+1}_{i+\frac{1}{2}} = f(S^{n+1}_{i+\frac{1}{2}}, S^{n+1}_{i+\frac{1}{2},L}) + \frac{2(C^I(S^{n+1}_{i+\frac{1}{2},L}) - C^I(S^{n+1}_i))}{\Delta x},
\]

and

\[
p_c^I(S^{n+1}_{i+\frac{1}{2},L}) = p_c^I(S^{n+1}_{i+\frac{1}{2},R}).
\]

Here, the superscripts \( I \) and \( II \) refer to the two different capillary-pressure functions.

In the presence of viscous and buoyancy forces, the phase-flow direction of 1D transport with constant total-velocity can be determined from the saturation information according to Eqn. 3.3. However, in the presence of capillary forces, the flow directions of the two phases cannot be determined based only on the saturation information. The pressure distribution, which obtained by solving the flow problem, is needed to calculate the directions of the two phases when capillary forces are significant. Hence, phase-based upstream weighting, which requires information about the phase-flow direction, cannot be used to calculate the capillary flux for 1D transport problem if the pressure field is not available. Without information about the flow directions of the fluid phases, central differencing of \( C(S) \) provides a good approximation for calculating capillary flux within a finite-volume discretization framework. (Numerical validation is provided in Appendix B.) From here on, we use central difference of \( C(S) \) to calculate the capillary flux for 1D transport in Eqn. 3.2

With the numerical discretization scheme fixed, we analyze the convergence behavior of Newton-based solvers. The Kantorovich theorem ([48]) provides a sufficient
condition for convergence of the Newton iterations. The theory [29, 30] states that if the residual $R(S)$ is a $C^2$ function of the solution $S$, Newton’s method is guaranteed to converge if

$$\frac{|R(S)R''(S)|}{|R'(S)|^2} < 1. \quad (3.7)$$

We define $\frac{|R(S)R''(S)|}{|R'(S)|^2}$ as the ‘convergence ratio’. This is a local result. In the vicinity of a root of $R$, the residual $|R(S)|$ is expected to be small, and the value of the convergence ratio is dominated by $|R'(S)|$. That is, when we are ‘close’ to the root, the convergence ratio is likely to be less one, and the iterations will converge quickly if $|R'(S)|$ is large. For the 1D transport problem, we need to find the argument (saturation) such that $R' = (R')_{\text{max}}$.

We consider immiscible two-phase flow in a single cell (control volume). Assume that $u_t$ is positive from left to right, and that the initial condition is $S^n = 0$. The saturations, $S_L$ and $S_R$, are the left and right boundary conditions. Then, the residual form of the hyperbolic conservation law can be written as

$$R = S^{n+1} + \frac{\Delta t}{\Delta x} (F_R^{n+1} - F_L^{n+1}), \quad (3.8)$$

where

$$F_R = f_R(S^{n+1}; S_R) + \frac{C(S_R) - C(S^{n+1})}{\Delta x}, \quad (3.9)$$

and

$$F_L = f_L(S_L; S^{n+1}) + \frac{C(S^{n+1}) - C(S_L)}{\Delta x}. \quad (3.10)$$

The first and second derivatives of the residual are

$$R' = \frac{dR}{dS^{n+1}} = 1 + \frac{\Delta t}{\Delta x} \left( \frac{\partial f_R(S^{n+1}; S_R)}{\partial S^{n+1}} - \frac{\partial f_L(S_L; S^{n+1})}{\partial S^{n+1}} - \frac{2}{\Delta x} \frac{dC(S^{n+1})}{dS^{n+1}} \right). \quad (3.11)$$
Next, we present a nonlinear solver for 1D multiphase transport in porous media that is unconditionally convergent, even in the presence of significant buoyancy and capillary forces. First, we discuss the nonlinear behavior in the absence of buoyancy and capillary forces. Then, we extend the solver to treat counter-current flow due to buoyancy and capillarity. Finally, we present a general transport solver to handle problems where viscous, buoyancy and capillary forces are all present and significant.

### 3.2 Cocurrent flow

In the absence of gravity and capillarity, \( f_R(S_n^{n+1}; S_R) = f_R(S_n^{n+1}) \), \( f_L(S_L; S_n^{n+1}) = f_L(S_L) \), and \( C = 0 \). Therefore, the residual can be written as

\[
R = S_n^{n+1} + \frac{\Delta t}{\Delta x} (f_R(S_n^{n+1}) - f_L(S_L)).
\]

(3.13)

It follows that

\[
R' = \frac{dR}{dS_n^{n+1}} = 1 + \frac{\Delta t}{\Delta x} \left( \frac{\partial f_R(S_n^{n+1})}{\partial S_n^{n+1}} \right),
\]

(3.14)

and

\[
R'' = \frac{d^2R}{d(S_n^{n+1})^2} = \frac{\Delta t}{\Delta x} \left( \frac{\partial^2 f_R(S_n^{n+1})}{\partial(S_n^{n+1})^2} \right).
\]

(3.15)

Since \( R \) is a \( C^2 \) function of \( S_n^{n+1} \), the necessary condition for \( R' = (R')_{\text{max}} \) is \( R'' = 0 \), which leads to

\[
\frac{d^2 f_R(S_n^{n+1})}{d(S_n^{n+1})^2} = 0.
\]

(3.16)

Note that the saturation, which satisfies Eqn. (3.16), is the inflection point of the fractional-flow curve used in the flux-based Newton method of Jenny et al. [12]. So, if there is no gravity, or capillarity, then in order to guarantee convergence, the solution...
from one Newton iteration to the next is not allowed to cross the inflection point of the flux function.

Consider a single-cell problem in the presence of viscous forces only. The boundary condition is set as \( S_L = 1 \) and \( S_R = 0 \). The relative permeabilities and viscosities are: \( k_{rw} = S^2 \), \( k_{rn} = (1 - S)^2 \), and \( \mu_w = \mu_n = 1 \). The inflection point of the flux function (Eqn. 3.2) is \( S = 0.5 \). The derivative of the residual with respect to the scalar saturation, \( R' \), and the convergence ratio \( \frac{|RR''|}{|R'|^2} \), respectively, are plotted versus saturation for different timestep sizes in Fig. 3.2. Note that \( c = \frac{\Delta t}{\Delta x} \), which is the total throughput (expressed in cell pore volumes), denotes the dimensionless timestep size. In Fig. 3.2(a) all the \( R' \) curves are smooth indicating \( C^2 \) continuity of the residual as a function of saturation. Large timestep sizes (e.g., \( c = 10 \)) correspond to strong nonlinearity. And from Fig. 3.2(a) it is clear that for different timestep sizes, the maxima of \( R' \) all occur at \( S = 0.5 \), i.e., the inflection point of the flux function. In Fig. 3.2(b) we can see that when the timestep is very small, e.g., \( c = 0.1 \), the convergence ratio is below unity for all saturation values. According to the Kantorovich theorem

\[ \text{Figure 3.2: Derivative of the residual and the convergence ratio for single-cell incompressible two-phase in the absence of gravity and capillarity: (a) } R' = \frac{dR}{dS}, \text{ (b) } \frac{|RR''|}{|R'|^2}. \]

\( c = \frac{\Delta t}{\Delta x} \), \( M^0 = 1 \). The inflection point of flux function is \( S_{\text{infl}} = 0.5 \).
convergence is guaranteed when the timestep is small enough. On the other hand, as the timestep size increases, the range of saturation values for which the convergence ratio is below unity gets smaller, and that implies potential convergence difficulties of Newton methods when the timestep size is large. Note that at $S = 0.5$, the convergence ratios are all zero (Fig. 3.2(b)), and there is a saturation region around $S = 0.5$, such that $\frac{|R R''|}{|R'|}$ is smaller than unity for all timesteps. Consequently, based on Kantorovich theorem, the nonlinear solution is guaranteed to converge once the iterative solution resides in the neighborhood of the inflection point ($S = 0.5$). That is, the inflection point is crucial to guarantee Newton convergence, which is consistent with the findings of Jenny et al. [12]. Based on this analysis, we can extend the modified Newton scheme of Jenny et al. [12] to cases with gravity and/or capillarity.

3.3 Counter-current flow due to gravity

When buoyancy is significant, counter-current flow may take place, i.e., the wetting and nonwetting phases have opposite flow directions at the same interface. In such problems, the unit-flux saturation, $S_f = 1$, plays a critical role in the solution process. This can be seen from Eqn. 3.3. Once the saturation crosses the unit-flux point, the flow direction of the nonwetting phase changes. That is, the upstream cell, which must be used to evaluate the nonwetting phase flux, flips sides. As a result, the residual, as a function of saturation, is not differentiable at the unit-flux point, and the assumption in the Kantorovich theorem regarding $C^2$ continuity is no longer valid. Nevertheless, on either side of the unit-flux point, the residual is $C^2$, and the Kantorovich theorem holds.

We study a single-cell problem with boundary conditions $S_L = 0.6$ and $S_R = 0.8$. $M^0 = 1$ and $N_g = -5$. In this setting, buoyancy forces are more dominant than the
viscous forces. The residual of our single-cell problem can be written as

\[ R = S^{n+1} + \frac{\Delta t}{\Delta x} \left( f_R(S^{n+1}; S_R) - f_L(S_L; S^{n+1}) \right). \]  
(3.17)

Fig. 3.3 shows the flux function, residual, derivative of the residual with respect to saturation, and the convergence ratio for this case. In Fig. 3.3(a), the unit-flux point of flux function is \( S^f = 1 = 0.447 \), and the two inflection points are \( S^{\text{inflec}_1} = 0.342 \) and \( S^{\text{inflec}_2} = 0.791 \).
is \( S_f^{1} = 0.447 \), and the sonic point is \( S_{\text{sonic}} = 0.630 \). There are two inflection points at \( S = 0.342 \) and \( S = 0.791 \). In Fig. 3.3(b) the residual is plotted versus saturation for different timesteps (i.e., \( c = \frac{\Delta t}{\Delta x} \)). It is clear that when the timestep sizes are large, the residual curves have strong nonlinearity. The intersection of the residual curves corresponds to the steady-state solution, i.e., the saturation value where the influx into the cell equals to the out-flux. The steady-state solution depends on the boundary conditions and is independent of timestep size. The solution is the saturation value such that \( R = 0 \). In Fig. 3.3(b) we can see that as the timestep size increases, the value of the saturation solution also increases. As the timestep size goes to infinity, the solution approaches the steady state. Notice that all the residual curves have a kink at the same saturation value, i.e., the unit-flux point \( S_f^{1} = 0.447 \). The non-smoothness of the residual curves at the unit-flux point is caused by the nonwetting phase flux at the right boundary flipping sides; hence, the out-flux is evaluated using the saturation solution in the cell itself and the right boundary condition. Thus, the derivative of the residual (Fig. 3.3(c)) changes significantly across the unit-flux point. Within \( S \in [0, S_f^{1}] \), \( R' \) is smooth and the local maximum of \( R' \) occurs at the inflection point \( (S = 0.342) \) irrespective of the dimensionless timestep size, \( c \). In Fig. 3.3(d) the convergence ratios versus saturation are shown for different timestep sizes. It is clear that when the timestep size is very small, e.g., \( c = 0.1 \), the convergence ratio is below unity. According to the Kantorovich theorem, the Newton method is unconditionally convergent when \( c = 0.1 \). On the other hand, the convergence ratios are not necessarily below unity when \( c \geq 0.5 \). Importantly, the convergence ratios for the different timestep sizes are all zero at the inflection point within \( S \in [0, S_f^{1}] \), i.e., \( S_{\text{infl}}^{1} \). So around \( S_{\text{infl}}^{1} \) convergence for the Newton iterations is guaranteed.

In order to construct an unconditionally convergent Newton-based scheme, we should not allow two successive saturation updates to reside on opposite sides of the unit-flux point. This ‘chopping’ of the update based on the unit-flux point is
necessary to overcome convergence difficulties due to flow reversal. Within each of
the two regions separated by the unit-flux point, we apply chopping based on the
inflection points in exactly the same manner as we do for cocurrent flow. In Fig. 3.3
the plots for $R$, $R'$, or the convergence ratio do not exhibit any special features due to
the sonic point. We employ phase-based upwinding, and it is not surprising that the
sonic point does not play a critical role in the nonlinear Newton-based solver. The
situation would be different, if, for example, a Godunov scheme is used.

3.4 Capillarity

When the transport is governed by both viscous and capillary forces, it is clear from
Eqn. 3.12 that $R'' = 0$ is equivalent to

$$\frac{d^2 f_R(S_{n+1})}{d(S_{n+1})^2} - \frac{2}{\Delta x} \frac{d^2 C(S_{n+1})}{d(S_{n+1})^2} = 0. \quad (3.18)$$

Then, the root of $R'' = 0$ is the solution of

$$\frac{d^2 F_R(S_{n+1})}{d(S_{n+1})^2} = 0, \quad (3.19)$$

where $F$ is defined as

$$F = f - \frac{2}{\Delta x} C. \quad (3.20)$$

Here, $f$ is the viscous flux, and $C$ is the $C(S)$ in Eqn. 3.2 representing the capillary
effect. In the absence of capillarity, $C$ vanishes, and $F$ equals $f$. The expression of
$F$ shows that when the grid size, $\Delta x$, is large, $F$ is close to $f$, and the capillary flux
can be ignored. That is, the significance of the capillary effects depends on the size
of the cells. Capillarity is expected to be important at relatively small scales.

Based on Eqn. 3.19 we employ chopping based on $F$ instead of $f$ in order to
update the solution around the region where $R' = (R')_{\text{max}}$. Thus, chopping based on the inflection point of $f$ in [12] is a special case of chopping based on $F$ when capillarity is ignored.

Fig. 3.4 shows the residual and convergence ratio for a single-cell problem with boundary conditions $S_L = 1$ and $S_R = 0$ when $M^0 = 1$ and $P_e = 0.2$. Fig. 3.4(b)

Figure 3.4: Convergence ratio for single-cell two-phase transport in the presence of viscosity and capillarity: (a) viscous flux function $f(S)$ (inflection point: 0.5); (b) total flux function $F(S)$ (inflection point: 0.383); (c) $R' = \frac{dR}{dS}$; (d) convergence ratio $\frac{|RR''|}{|R'|}$. $c = \frac{\Delta t}{\Delta x}$. $M^0 = 1$, $P_e = 0.2$. 
shows that the flux function $F(S)$ has a similar S shape to that of $f(S)$. For this case, the inflection point of $F$ is $S = 0.383$, while the inflection point of $f$ is $S = 0.5$. In Figs. 3.4(c) and 3.4(d), all the maxima of the residual derivatives occur at $S = 0.383$, which is the inflection point of $F$, and all the convergence ratios are zero at that point. Note that in Fig. 3.4(d) each curve has two zeros. The convergence ratio, $\frac{\left|R''R\right|}{\left|R'\right|^2}$, is zero when either $R = 0$ or $R'' = 0$. Thus, the two zeros in each convergence ratio curve, respectively, represent the inflection point of $F$, where $R'' = 0$, and the solution of the single-cell problem, where $R = 0$.

### 3.5 General transport scheme

Before describing the general transport scheme, we first introduce the concept of ‘trust region’ in the context of nonlinear solvers. If the entire range of the unknown can be divided into several subregions, such that the convergence of the nonlinear solver is guaranteed once the iterative solutions are confined in the same subregion as the true solution, these subregions are called ‘trust regions’ [30]. Here, trust-region Newton methods refer to a specific type of Newton methods that partitions the space into trust regions to guide the Newton update in order to improve the convergence behavior. Based on this concept, we now describe a nonlinear solution strategy for multiphase transport problems with viscous, buoyancy, and capillary forces. Essentially, it is a cell-wise chopping strategy guided by trust regions of the flux function. The flux function is divided into saturation trust regions. The delineation of these regions is dictated by the inflection of $F$ (i.e., $S_{\text{inflec}}$), the unit-flux point of $f$, $S^{f=1}$, and the end points. Our trust-region Newton method ensures that two successive iterations cannot cross any trust-region boundary. If a crossing is detected, we 'chop back' the saturation value at the appropriate trust-region boundary. The algorithm is summarized in Fig. 3.5.
Figure 3.5: The flow chart of the trust-region Newton scheme in the presence of viscous, gravitational and capillary forces, for one timestep.
3.6 Global convergence of multiphase transport

In this section, we provide proof of global convergence of our trust-region Newton method for single-cell problems. First, we need to define the terms ‘local’ and ‘global’. ‘Local’ Newton methods refer to the situation that ‘sufficiently good’ initial guesses of the solution are assumed to be at hand ([30]). One example is the ordinary Newton method. Local Newton methods are not guaranteed to converge. ‘Global’ Newton methods refer to Newton-like methods whose convergence is independent of the initial guess and timestep size. For Newton methods, convergence of the nonlinear transport problem is guaranteed only if the global structure of residual function, \( R = R(S) \), is exploited. Hence, the chopping strategy (as shown in Fig. 3.5) in our trust-region Newton method is a ‘globalization’ technique for the ordinary Newton method. Now we prove how our trust-region method guarantees the convergence and thus ‘globalizes’ the Newton method for our class of problems.

A critical piece of the analysis is that the direction of the Newton update is always toward the solution. The statement is summarized in lemma 3.6.1.

**Lemma 3.6.1.** (Local Descent) Let \( R : \mathbb{D} \rightarrow \mathbb{R}^n \) be a piecewise continuously differentiable mapping with \( \mathbb{D} \subseteq \mathbb{R}^n \). That is, \( \mathbb{D} = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \cdots \cup \mathbb{D}_k \), where \( \mathbb{D}_i \) \((i = 1, k)\) is open domain and \( R \in C^1(\mathbb{D}_i) \). Define the level set of \( R \) as

\[
T(x) := \frac{1}{2} ||R(x)||_2^2 = \frac{1}{2} R(x)^T R(x).
\]  

(3.21)

Let \( R(x) \neq 0 \) and \( \Delta x \neq 0 \). Then there exists a \( \mu > 0 \), such that \( T(x + s\Delta x) < T(x) \), \( 0 < s < \mu \).

The proof of this lemma can be found Appendix [C] which is an extension for the global convergence of continuously differentiable functions in [30]. Based on this lemma, the following theorem, which relates to global convergence of our trust-region
Newton method, can be proved.

**Theorem 3.6.1.** For a single-cell transport problem, the residual, \( R = R(S) \), is a piecewise \( C^1 \) function of saturation. Starting from an arbitrary initial guess \( S^{n+1,0} \in [0, S^{f=1}) \cup (S^{f=1}, 1] \), the trust-region Newton converges to the solution point \( S^{n+1} \).

**Proof.** The saturation space can be partitioned into trust regions. The delineation of these regions is dictated by the inflection, unit-flux, and end-points. That is,

- If \( S^{f=1} > S^{inflec1} \), the trust regions are
  \[
  [0, S^{inflec1}] \cup [S^{inflec1}, S^{f=1}] \cup (S^{f=1}, S^{inflec2}] \cup [S^{inflec2}, 1].
  \]

- If \( S^{f=1} < S^{inflec1} \), the trust regions are
  \[
  [0, S^{f=1}) \cup (S^{f=1}, S^{inflec1}] \cup [S^{inflec1}, S^{inflec2}] \cup [S^{inflec2}, 1].
  \]

We focus on the proof for \( S^{f=1} > S^{inflec1} \). The proof for the case with \( S^{f=1} < S^{inflec1} \) can be obtained in a similar manner.

For \( S^{f=1} > S^{inflec1} \), typical flux functions are shown in Fig. 1.1. It is observed that within each trust region, the flux functions, and hence the residual functions, are \( C^1 \) and convex (or concave).

If the initial guess, \( S^{n+1,0} \), and the solution, \( S^{n+1} \), reside in the same trust region, convergence is guaranteed. This is because the trust-region Newton scheme will limit the iterative solutions to stay within this trust region, and the Newton iterative solutions will converge globally for convex (or concave) residual function (30). On the other hand, if the initial guess, \( S^{n+1,0} \), and the solution, \( S^{n+1} \), reside in different trust regions of the flux, we prove global convergence of our trust-region Newton method with the following road map.
First, Lemma 3.6.1 tells us that Newton updates always provide the correct direction, i.e., in each iteration, the iterative solution always jumps from its current value ‘towards’ the solution \[30\]. Second, if the timestep size is small, the next update may reside in the same trust region as the current solution and jump towards the solution monotonically. On the contrary, if the timestep is large, the next iterative solution may cross one, or multiple, trust regions from the current solution. The trust-region Newton method forces the iterative solutions to stop at all the detected boundaries, i.e., \(S_{\text{inter}}\) or \(S_f=1\). Therefore, the iterative solution can jump to the solution monotonically in terms of trust regions, i.e., from the current trust region to a trust region that is closer to the solution, or that contains the solution. Third, once the iterative solution jumps inside the trust region containing the solution, the residual curve is convex/concave, and hence convergence is guaranteed. Note that within the trust region containing the solution, the Newton iterative solution does not necessarily proceed towards the solution monotonically. But, since within each trust region, we only need the standard local Newton method, convergence is achieved.

\[\square\]

### 3.7 Numerical examples

The performance of our trust-region Newton method is demonstrated using numerical examples. Starting with very simple single-cell transport problems, we use convergence maps (12) to investigate the behavior of the nonlinear solver for arbitrary combinations of the boundary conditions and the initial guesses. Then, we investigate the so-called Newton flow (11) and the Newton iterations for two-cell transport problems. Following the two-cell problems, 1D nonlinear transport problems with multiple cells are studied. We look at how much computational effort we can save by using trust-region Newton, if convergence is the only criterion of simulation and accuracy is not the main concern. Finally, we test trust-region Newton for the upper
zone of the SPE 10 model (Tarbert formation) and the full SPE 10 models. For all the test cases, we compare the convergence performance between standard Newton and our trust-region Newton method.

3.7.1 Single-cell problems: convergence maps

Nonlinear analysis of single-cell transport problems reveals some fundamental conclusions that are valid for multiple cells and even large-scale problems. Consider the single-cell case problem shown in Fig. 3.6. Given boundary conditions on the left and right sides ($S_L$ and $S_R$), starting from initial guess $S^{n+1,0}$, the objective is to find the solution $S^{n+1}$ of the nonlinear transport problem (Eqn. 1.22).

The behaviors of the standard Newton scheme and our trust-region Newton scheme are shown using convergence maps in Fig. 3.7 and Fig. 3.8 respectively. The colors in Fig. 3.7 and 3.8 depict the number of iterations required until the convergence criteria $|S^{n+1,\nu+1} - S^{n+1,\nu}| < 10^{-5}$ and $|R| < 10^{-4}$ are satisfied. Specifically, dark blue means
fast convergence (1 iteration) and dark red means slow convergence (15 iterations); the white region in the phase space $S_{n+1}^{0} - S^{L}$ indicates that no convergence is achieved after 200 iterations.

\[ (a) \quad c = 0.1 \]
\[ (b) \quad c = 1 \]
\[ (c) \quad c = 10 \]
\[ (d) \quad c = 100 \]

Figure 3.7: Convergence map of standard Newton scheme for single-cell transport with viscous flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$. $S_R = 1$, $M^0 = 1$, and $N_g = 0$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).

For this case, where $M^0 = 1$, the flux function is shown in Fig. 1.1(a). There is one inflection point $S^{\text{inflec}} = 0.5$. Fig. 3.7 shows that for small timesteps ($c = 0.1, 1$), the standard Newton scheme is convergent for any initial guess, i.e., the convergence map contains no white regions. However, as the timestep size increases ($c = 10, 100$),
convergence of the standard Newton method is no longer guaranteed. Notice that the size of the non-convergence (white) area increases as the timestep size increases. Another observation is that when the timestep is quite large ($c = 100$), the convergence map in Fig. 3.7(d) is almost symmetric (it is perfectly symmetric when the timestep size is infinite). The center is $(S^{n+1,0}, S_L) = (0.5, 0.5)$, which is the inflection point of the flux function.

Figure 3.8: Convergence map of trust-region Newton scheme for single-cell transport with viscous flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$, $S_R = 1$, $M^0 = 1$, and $N_g = 0$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
On the other hand, Fig. 3.8 shows that the trust-region Newton scheme is convergent for all the timestep sizes, i.e., there are no white regions for any timestep size. Although as the timestep increases, more Newton iterations are required, the number of iterations is small, i.e., below 10 even for very large timesteps ($c = 100$). In the absence of gravity and capillarity, our trust-region Newton scheme is identical to the modified Newton method described by Jenny et al. [12]. That is, we apply chopping at the inflection point of the flux function. In this case, the inflection point is $S^{\text{infl}} = 0.5$. It is observed that for $c = 100$, there is a jump in the convergence behavior (represented by color in Fig. 3.8(d)) across $S^{n+1.0} = 0.5$. This is due to chopping at $S = 0.5$.

Viscous and gravitational forces

Next, we study the behavior for cases with both viscous and buoyancy forces. Specifically, we set $N_g = -5$, for which buoyancy is more dominant than the viscous forces. The flux function is shown in Fig. 1.1(b). Again, we fix the right boundary condition as $S_R = 1$, and we study if the root of the residual (Eqn. 3.13) is found by Newton method as $\nu \to \infty$ for all possible starting points ($S^{n+1.0}, S_L \in (0, 1) \times (0, 1)$). The behaviors of the standard Newton scheme and the proposed trust-region Newton scheme are shown using convergence maps in Fig. 3.9 and Fig. 3.10 respectively.

Fig. 3.9 shows that the white region is larger when the timestep size $c$ increases. In this case, there is no guarantee for the standard Newton to converge when $c \geq 1$. Moreover, the convergence behavior (reflected by the color in the convergence maps) is quite different for $S_L < S^{f=1}$ and $S_L > S^{f=1}$, as well as, for $S^{n+1.0} < S^{f=1}$ and $S^{n+1.0} > S^{f=1}$, where $S^{f=1} = 0.447$ is the unit-flux point on the flux function. Thus, the unit-flux point plays a critical role in the convergence behavior.

We apply chopping based on the unit-flux point, as well as, the inflection points in our trust-region Newton method. Fig. 3.10 indicates that convergence is achieved
Figure 3.9: Convergence map of standard Newton scheme for single-cell transport with viscous and gravitational flux: (a) \( c = 0.1 \); (b) \( c = 1 \); (c) \( c = 10 \); (d) \( c = 100 \). The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
Figure 3.10: Convergence map of trust-region Newton scheme for single-cell transport with viscous and gravitational flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$, $S_R = 1$, $M^0 = 1$, and $N_g = -5$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
for all the timesteps (i.e., no white regions in the maps). For any timestep size, the trust-region scheme converges for arbitrary combinations of the left boundary condition and the initial guess. In Fig. 3.10 the number of iterations used to achieve convergence (reflected by the color) has discontinuities across the two inflection points ($S_{\text{inflec}} = 0.342$ and 0.791) and one unit-flux point ($S_{f=1} = 0.447$) on Fig. 1.1(b) corresponding to the chopping strategy applied at these points.

Figure 3.11: Convergence map of Newton scheme with Appleyard chopping for single-cell transport with viscous and gravitational flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$. $S_R = 1$, $M^0 = 1$, and $N_g = -5$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).

To complete the comparison, we also plot the convergence map for the Newton’s
method with the Appleyard chopping, which is used in the Eclipse simulator [49, 52] (Fig. 3.11). Appleyard chopping is a heuristic scheme that limits the saturation change per iteration to be less than a fixed value (e.g., 0.2). It is observed that for this case, the Newton method with Appleyard chopping converges for arbitrary initial guesses and boundary conditions when \( c = 0.1, 1 \). When \( c \geq 10 \), convergence of the Newton method with Appleyard chopping is no longer guaranteed, as shown by the small white regions in the convergence map.

If buoyancy effects are even more dominant, say, \( N_g = -20 \), the convergence behavior of trust-region Newton is superior to both the standard Newton method and the Newton method with Appleyard chopping. As shown in Fig. 3.12 and 3.14, the standard Newton scheme and the Newton method with Appleyard chopping are convergent for every point of the phase plane \( S^{n+1,0} - S_L \), only when the timestep size is very small, e.g., \( c = 0.1 \), and there are non-convergent points (white region) in the phase plane when \( c = \{1, 10, 100\} \). In contrast, Fig. 3.13 shows that the trust-region scheme is convergent for any timestep size. In Fig. 3.13, the discontinuities in the number of iterations (reflected by the color) correspond to chopping at the two infection points (\( S^{inflec} = 0.298 \) and 0.737) and the unit-flux point (\( S^{f=1} = 0.224 \)) when \( N_g = -20 \). Chopping at these points resolves the convergence difficulty of the nonlinear solver when the timestep is large. In fact, Newton with Appleyard chopping also applies chopping when saturation update is larger than a pre-specified value, e.g., 0.2. Since 0.2 is only a heuristic estimation of the maximum allowable Newton update and the chopping is usually not applied at the trust-region boundaries, Appleyard chopping may improve convergence, but not always fully resolve the nonlinearity of the flux function and ensure unconditional convergence.
Figure 3.12: Convergence map of standard Newton scheme for single-cell transport with viscous and gravitational flux: (a) \( c = 0.1 \); (b) \( c = 1 \); (c) \( c = 10 \); (d) \( c = 100 \). \( c = \frac{\Delta t}{\Delta x^2} \), \( S_R = 1 \), \( M^0 = 1 \), and \( N_g = -20 \). The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
Figure 3.13: Convergence map of trust-region Newton scheme for single-cell transport with viscous and gravitational flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$, $S_R = 1$, $M^0 = 1$, and $N_g = -20$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
Figure 3.14: Convergence map of Newton scheme with Appleyard chopping for single-cell transport with viscous and gravitational flux: (a) \( c = 0.1 \); (b) \( c = 1 \); (c) \( c = 10 \); (d) \( c = 100 \). \( c = \frac{\Delta t}{\Delta x} \). \( S_R = 1 \), \( M^0 = 1 \), and \( N_g = -20 \). The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
CHAPTER 3. TRUST-REGION NEWTON FOR TRANSPORT

Viscous and capillary forces

For the single-cell transport problem with viscous and capillary forces, we set $P_e = 0.2$, which indicates that both the viscous and capillary effects are quite important. For this case, the flux function, $F$, has one inflection point, $S^{inflec} = 0.383$, as shown in Fig. 1.1(c). We study the behavior of standard Newton and trust-region Newton methods, respectively, using convergence maps. Fig. 3.15 shows that the standard Newton scheme is convergent for every point of the phase plane $S^{n+1,0} - S_L$, only when the timestep size is very small, e.g., $c = 0.1$. And there are non-convergent points (white region) in the domain when $c = \{1, 10, 100\}$. The size of the non-convergent region increases as the timestep is increased. On the contrary, by applying chopping on the inflection point of $F$, the trust-region Newton scheme is convergent for any timestep size (as shown in Fig. 3.16). The discontinuity of the number of iterations (reflected by colors) at $S^{n+1,0} = 0.383$ in Fig. 3.16 is a reflection of the chopping scheme in our trust-region method.

Viscous, gravitational, and capillary forces

The objective of our trust-region Newton is to resolve the nonlinearity for immiscible two-phase transport across the entire viscous-buoyancy-capillary parameter space. Now, we investigate the convergence behaviors of standard and trust-region Newton schemes in the presence of viscous forces, strong buoyancy ($N_g = -5$), and strong capillarity ($P_e = 0.2$). For the trust-region Newton method, we apply chopping at the unit-flux point of $f$ (viscous and buoyancy flux function) and inflection points of $F$ (viscous, buoyancy and capillary flux function). For the standard Newton method, the convergence maps for various timestep sizes ($c = 0.1, c = 1, c = 10$ and $c = 100$) are shown in Figs. 3.17. We can see that when the timestep size is small ($c = 0.1$), there is no white region in the convergence map. The figure shows that when
Figure 3.15: Convergence map of standard Newton scheme for single-cell transport with viscous and capillary flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$. $M^0 = 1$, and $P_e = 0.2$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
Figure 3.16: Convergence map of trust-region Newton scheme for single-cell transport with viscous and capillary flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$, $M^0 = 1$, and $P_e = 0.2$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
c = 0.1, the standard Newton method is unconditionally convergent, i.e., given any boundary condition, convergence is achieved starting from any initial guess. As the timestep size increases, the number of Newton iterations increases and there are white (unconvergent) regions in the convergence maps when $c \geq 1$. That is, for the standard Newton method, convergence is not guaranteed when $c \geq 1$. In fact, we can see that for any given boundary condition (y-axis), standard Newton is not convergent (represented by white regions in the convergence maps) for some initial guesses (x-axis). In Fig. 3.17, it is clear that there are discontinuities in the convergence maps at $S^{n+1,0} = S^{f=-1}$. The discontinuity is introduced by flow reversal and becomes obvious when the timestep size is large ($c = 10$, and 100).

Fig. 3.18 displays the convergence maps for our trust-region Newton method. There are no white regions in the convergence maps for any timestep size, which indicate that the trust-region Newton method is unconditionally convergent. The number of Newton iterations does increase slightly with timestep size for most combinations of the boundary condition and the initial guess. In Fig. 3.18, there are discontinuities in the convergence maps at $S^{n+1,0} = S^{\text{inflc1}}$, $S^{n+1,0} = S^{f=-1}$, and $S^{n+1,0} = S^{\text{inflc2}}$. The discontinuities are introduced by the chopping at these trust-region boundaries. We can see that the chopping makes this ‘modified’ Newton method ‘globally’ convergent for single-cell transport problems.

### 3.7.2 Two-cells transport: Newton flow

Now, we study two-cell transport problems to illustrate how the chopping scheme in the trust-region Newton method guides the Newton iteration toward the solution and improves convergence substantially. As shown in Fig. 3.19, the boundary conditions are provided on the left and right sides for these two-cell problems.
Figure 3.17: Convergence map of standard Newton scheme for single-cell transport with viscous, gravitational and capillary flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$. $M^0 = 1$, $N_g = -5$, and $P_e = 0.2$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).
Figure 3.18: Convergence map of trust-region Newton scheme for single-cell transport with viscous, gravitational and capillary flux: (a) $c = 0.1$; (b) $c = 1$; (c) $c = 10$; (d) $c = 100$. $c = \frac{\Delta t}{\Delta x}$. $M^0 = 1$, $N_g = -5$, and $P_e = 0.2$. The color refers to the convergence rate of the scheme: dark blue means fast (1 iteration) and dark red slow (15 iterations) convergence; white indicates no convergence (not convergent within 200 iterations).

Figure 3.19: Two-cell transport problem with left and right boundary conditions
CHAPTER 3. TRUST-REGION NEWTON FOR TRANSPORT

Viscous forces

First, we study two-cell transport with only viscous forces. We set $M^0 = 1$, and we apply unit saturation as the left boundary condition, i.e., $S_L = 1.0$, and right boundary is $S_R = 0.0$. The initial condition is $S = [0.0, 0.0]$ and the timestep is $\Delta t = 10$. We plot the residual norm contours, the Newton flow, as well as, the iterative solutions in Fig. 3.20, respectively, for standard Newton and the trust-region scheme. The Newton flows are obtained using a high-fidelity numerical integrator (variable-order Runge-Kutta with automatic step control [50]), and they illustrate the continuous paths that Newton’s method attempts to approximate should it have been started from these various initial guesses (111).

In Fig. 3.20, the contour lines on the figures represent the 2-norm of the residual in the two-dimensional saturation space, $S \in [0, 1] \times [0, 1]$. The solution, $S^{n+1} = [0.7753; 0.7057]$, is the point with the smallest residual norm. The thick, curved black lines are Newton flows emanating from various starting points on the boundary of the $(S_1, S_2)$ (2D) space. All the Newton flows have the same end point, which is the solution. On the other hand, the thick color lines in Fig. 3.20 represent the Newton iterative solutions.

We observe that if a point close to the solution is chosen as the initial guess, the standard Newton method is convergent. However, if we choose the initial condition as the initial guess, the iterative solutions from the standard Newton jump between the corner points on $S \in [0, 1] \times [0, 1]$ and do not converge (Fig. 3.20(a)). On the contrary, we observe that the trust-region Newton is convergent from arbitrary starting points. In Fig. 3.20(b) the thick colored lines illustrate the sequence of updates associated with the trust-region Newton method with several different initial guesses:

$S = [0.0, 0.0], [0.5, 0.0], [1.0, 0.0], [1.0, 0.5], [1.0, 1.0], [0.5, 1.0], [0.0, 1.0], \text{ and } [0.0, 0.5]$. 
Figure 3.20: Newton flow for two-cell transport with viscous flux. $N_g = 0$, $P_e = \infty$, $M^0 = 1$, $\Delta t = 10$, $S_L = 1.0$, and $S_R = 0.0$. The initial guess is $S_1 = 0.0$ and $S_2 = 0.0$. $[S_1^{n+1}, S_2^{n+1}] = [0.78, 0.71]$. (a) standard Newton scheme; (b) trust-region Newton scheme. In (a), $[S_1^n, S_2^n] = [0.0, 0.0]$; while in (b), Newton iterative solutions starting from multiple initial guesses $[S_1^n, S_2^n]$ are displayed.
All the sequences are convergent. Fig. 3.20(b) indicates our chopping strategy restricts
the iterative solutions to not jump too far in any given direction. As a result, the
residual norm gets progressively smaller with iteration until we land in the trust
region where the solution resides. We also show the numerical values of the iterative
solutions in Table 3.1 for the initial guess \( S = [0.0, 0.0] \). The results indicate that
the trust-region Newton converges after five iterations, while the standard Newton
oscillates between the end points and does not converge.

Table 3.1: Comparison of standard Newton versus our trust-region Newton scheme
for iterative solution of a two-cell transport with viscous flux \( (N_g = 0, P_e = \infty) \) with
\( M^0 = 1, \Delta t = 10, S_L = 1.0, \) and \( S_R = 0.0. \) The initial guess is \( S_1 = 0.0 \) and \( S_2 = 0.0. \)
\([S_1^n, S_2^n] = [0.0, 0.0]. \) The solution is \([S_1^{n+1}, S_2^{n+1}] = [0.78, 0.71]. \) The iteration results
\([S_1, S_2]\) are shown before and after ‘chopping’.

<table>
<thead>
<tr>
<th>#iter</th>
<th>standard Newton</th>
<th>trust-region scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>before chopping</td>
<td>after chopping</td>
</tr>
<tr>
<td>1</td>
<td>[10, 0]</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>2</td>
<td>[0, 10]</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>3</td>
<td>[10, -10]</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>4</td>
<td>[0, 10]</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>5</td>
<td>[10, -10]</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>6</td>
<td>[0, 10]</td>
<td>[0, 1]</td>
</tr>
</tbody>
</table>

Viscous and gravitational forces

Now, we move to two-cell transport under both viscous and buoyancy forces. We
use the same initial and boundary conditions as the previous case. Let \( N_g = -5, \)
so the flux function is the one in Fig. 1.1(b). The unit-flux point is \( S = 0.447. \)
We simulate this case using both the standard Newton method and the trust-region
Newton for \( \Delta t = 10. \) Fig. 3.21 shows the residual contours, Newton flow, and the
iterative solutions, respectively, for two simulations.
When gravity is significant, the Newton flow curves are not smooth, and for saturation loci where the upwind directions change ($S = 0.447$), there are clear kinks on the Newton flows along the $S_1 = 0.447$ and $S_2 = 0.447$ (as the dashed orange lines plotted in Fig. 3.21(a)). Another observation is that the Newton flows emanating from some starting points flow beyond the region $[0, 1] \times [0, 1]$ and then flow back to the region, approaching the solution. The solution for this case is $S^{n+1} = [0.6104, 0.6907]$. It reveals that Newton method, by shooting along the local direction of Newton flows suffers from convergence difficulties when buoyancy is significant.

The thick color lines in Fig. 3.21 show the iterative solutions from Newton methods. In Fig. 3.21(a) using $S = [0.0, 0.0]$ as the initial guess, the standard Newton method oscillates between the end points and does not converge. On the other hand, the trust-region Newton converges quickly from an arbitrary initial guess. Fig. 3.21(b) illustrates the sequence of updates for several different initial guesses:

$$S = [0.0, 0.0], [0.5, 0.0], [1.0, 0.0], [1.0, 0.5], [1.0, 1.0], [0.5, 1.0], [0.0, 1.0], \text{ and } [0.0, 0.5].$$

All of the trust-region sequences are convergent and march toward the solution monotonically.

Table 3.2 provides the numerical values of the iterative solutions for both Newton methods with initial guess $S = [0.0, 0.0]$. The trust-region Newton converges within seven iterations, while the standard Newton method does not converge.

### 3.7.3 1D transport

We study two-phase transport in a one dimensional model with 80 cells (control-volumes) for various combinations of viscous, buoyancy and capillary forces. For each combination, we compare the convergence performance between the standard Newton and our trust-region Newton algorithm for several timestep sizes. We use the
\( S_1 \) \( S_2 \)

0 0.2 0.4 0.6 0.8 1

0 0.2 0.4 0.6 0.8 1

(a) standard Newton scheme

(b) trust-region Newton scheme

Figure 3.21: Newton flow for two-cell transport with viscous and gravitational flux. \( N_g = -5, P_e = \infty, M^0 = 1, \Delta t = 10, S_L = 0.6, \) and \( S_R = 0.8. \) The initial guess is \( S_1 = 0.0 \) and \( S_2 = 0.0. \) \( [S_1^n, S_2^n] = [0.61, 0.69]. \) (a) standard Newton scheme; (b) trust-region Newton scheme. In (a), \( [S_1^n, S_2^n] = [0.0, 0.0]; \) while in (b), Newton iterative solutions starting from multiple initial guesses \( [S_1^n, S_2^n] \) are displayed.
Table 3.2: Comparison of standard Newton versus our trust-region Newton scheme for iterative solution of a two-cell transport with viscous and gravitational flux \(N_g = -5, Pe = \infty\) with \(M^0 = 1, \Delta t = 10, S_L = 0.6,\) and \(S_R = 0.8\). The initial guess is \(S_1 = 0.0\) and \(S_2 = 0.0\). \([S_1^n, S_2^n] = [0.0, 0.0]\). The solution is \([S_1^{n+1}, S_2^{n+1}] = [0.61, 0.69]\). The iteration results \([S_1, S_2]\) are shown before and after ‘chopping’.

<table>
<thead>
<tr>
<th>#iter</th>
<th>standard Newton</th>
<th>trust-region scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>before chopping</td>
<td>after chopping</td>
</tr>
<tr>
<td>1</td>
<td>3.86, 0.00</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>2</td>
<td>4.12, 5.89</td>
<td>[1, 1]</td>
</tr>
<tr>
<td>3</td>
<td>0.00, -0.34</td>
<td>[0, 0]</td>
</tr>
<tr>
<td>4</td>
<td>3.86, 0.00</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>5</td>
<td>4.12, 5.88</td>
<td>[1, 1]</td>
</tr>
<tr>
<td>6</td>
<td>0.00, -0.34</td>
<td>[0, 0]</td>
</tr>
<tr>
<td>7</td>
<td>3.86, 0.00</td>
<td>[1, 0]</td>
</tr>
<tr>
<td>8</td>
<td>4.12, 5.88</td>
<td>[1, 1]</td>
</tr>
</tbody>
</table>

CFL number to quantify the timestep size. The CFL number for two-phase (wetting and nonwetting) flow is

\[
CFL = \frac{\Delta t \frac{\lambda}{\lambda_n} \lambda' u_n - \frac{\lambda}{\lambda_w} \lambda' u_w + \lambda_w \lambda_n k p'_c}{\phi L \frac{1}{\lambda_w + \lambda_n}}, \tag{3.22}
\]

where \(\Delta t\) is the dimensional timestep size and \(L\) the cell size. \(\lambda\) and \(\lambda'\) are the phase mobility and its derivative with respect to saturation. In the absence of capillary forces, the definition of the CFL number becomes

\[
CFL = \frac{\Delta t \frac{\lambda}{\lambda_n} \lambda' u_n - \frac{\lambda}{\lambda_w} \lambda' u_w}{\phi L \frac{1}{\lambda_w + \lambda_n}}. \tag{3.23}
\]

If buoyancy is ignored

\[
\begin{align*}
  u_w &= u_t \frac{\lambda}{\lambda_w + \lambda_n} \\
  u_n &= u_t \frac{\lambda}{\lambda_w + \lambda_n}
\end{align*}
\tag{3.24}
Hence, the CFL number is

\[ CFL = \frac{\Delta t u_t \lambda_w \lambda'_n - \lambda_n \lambda'_w}{\phi L \lambda (\lambda_w + \lambda_n)^2} \]

\[ = \frac{\Delta t u_t (\lambda_w + \lambda_n) \lambda'_n - \lambda_n (\lambda_w + \lambda'_n)'}{\phi L (\lambda_w + \lambda_n)^2} \]

\[ = \frac{\Delta t u_t \left( \frac{\lambda_n}{\lambda_w + \lambda_n} \right)'}{\phi L dS}. \]  

(3.25)

In [5], the CFL number is defined using dimensional quantities, as shown in Eqns. 3.22 and 3.25. By using \( \Delta t \) is the dimensionless timestep (i.e., cell pore-volume injected) as defined in Eqn. 1.19, we obtain

\[ CFL = \Delta t \frac{df_n}{dS}. \]  

(3.26)

**Viscous forces**

We study 1D transport under viscous forces only. The initial condition is \( S = 0.0 \) in the domain. We inject the wetting phase at the left boundary and produce at a constant rate from the right boundary. We solve the problem from the initial time of zero to a time of 5000 CPVI (cell pore-volume injected). After such a long time, the nonwetting phase in the 1D domain is fully displaced by the injected wetting phase. In Fig. 3.22, we compare the solution at 5000 CPVI obtained using \( \Delta t = 0.1 \) (CFL≈1.82) with the solution obtained using a single timestep \( \Delta t = 5000 \) (CFL≈1473). The CFL number is defined in Eqn. 3.26. From the numerical solutions in Fig. 3.22, it is clear that the larger timesteps are associated with larger amounts of numerical diffusion.

In Fig. 3.23(a) and 3.23(b), the average number of iterations per timestep is plotted versus the timestep size and the maximum CFL number for the standard Newton and trust-region Newton methods, respectively. Standard Newton can converge for \( \Delta t \leq \ldots \)
1 (CFL≤2.79), while the trust-region Newton is convergent even with \( \Delta t = 5000 \) (CFL≈1473). For cases when both solvers converge, although the average number of iterations per timestep increases as the timestep size increases, the total number of iterations decreases as the timestep size increases. For example, the trust-region Newton is convergent with \( \Delta t = 5000 \), and it only takes 11 total iterations to achieve convergence. For \( \Delta t \leq 1 \) where standard Newton can converge, the number of iterations per timestep is similar for the two solvers. In Fig. 3.23(c), we compare the total number of iterations between the two nonlinear solvers, each using its largest convergent timestep size: standard Newton with \( \Delta t = 1 \) and trust-region Newton with \( \Delta t = 5000 \). For such a setting, we can reduce the number of total iterations by almost three orders of magnitude by applying our trust-region scheme.
Figure 3.23: Performance comparison between standard and trust-region Newton schemes with the number of iterations required to solve a case of 1D incompressible two-phase Buckley-Leverett displacing problem under only viscous forces ($M^0 = 1$, $N_g = 0$ and $P_e = \infty$): (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations ($\Delta t = 1$ for standard Newton and $\Delta t = 5000$ for trust-region scheme).
Gravity segregation

We consider gravity segregation in a one-dimensional domain. The domain boundaries are closed and buoyancy is the only driving force. Initially, the top half of the domain ([1,40]) is wetting (taken as the phase heavier) phase and the bottom half of the domain ([41,80]) is nonwetting (lighter) phase. The flux function, as shown in Fig. 3.24(a), is no longer S-shaped. There are two inflection points and one sonic point ($S_{sonic} = 0.5$). There is no unit-flux point. For this gravity segregation case, the wetting and nonwetting phases are always flowing in opposite directions. The absence of a unit-flux point reveals that there is no chance for flow reversal, i.e., switching from counter-current flow to cocurrent flow.

Starting with the initial condition shown in Fig. 3.24(b), the phase-flow directions are fixed in this case. Specifically, the wetting phase moves downwards and the nonwetting phase moves upward. The saturation profiles at 5000 CPVI using $\Delta t = 0.1$ and $\Delta t = 5000$ are presented in Fig. 3.24(c). Note that while the saturation field appears more dispersed for the $\Delta t = 5000$ case due to the large time truncation errors, the solution is consistent with the discrete representation of the physics and is free of oscillations. Therefore, without worrying about the convergence of the nonlinear solver, the trust-region scheme allows us to choose the timestep size solely on accuracy considerations (i.e., space and time truncation errors).

Fig. 3.25(a) and 3.25(b) show the number of iterations per timestep versus the timestep size and the CFL number, respectively. The trust-region scheme is convergent for any timestep size, even when the CFL number is extremely large (1884 here). It is observed that the standard Newton scheme converges only for $\Delta t \leq 1$ (CFL number is around 3.8). Once the standard Newton scheme is able to converge, the number of iterations it needs is comparable to that for the trust-region scheme. Fig. 3.25(c) compares the total iterations needed for both schemes with the largest convergent timestep. The results indicate that compared with the standard Newton
Figure 3.24: Numerical results with trust-region Newton scheme of a 1D incompressible two-phase gravity segregation ($M^0 = 1$, $N_g = -\infty$ and $P_e = \infty$): (a) flux function (Red dots represent inflection points; and green dots are sonic points); (b) initial condition; (c) final solution with $\Delta t = 0.1$ (CFL$\approx$1.96) and with $\Delta t = 5000$ (CFL$\approx$1885).
Figure 3.25: Performance comparison between standard and trust-region Newton schemes with the number of iterations required to solve a case of 1D incompressible two-phase gravity segregation ($M^0 = 1$, $N_g = -\infty$ and $P_e = \infty$): (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations ($\Delta t = 1$ for standard Newton and $\Delta t = 5000$ for trust-region scheme).
method, we can save a significant amount of iterations, (i.e., computational effort) to obtain a convergent solution by using the trust-region scheme.

Capillary diffusion

We study the cases where capillary forces are the only driving forces. For this case, starting from a non-equilibrium initial condition (wetting in the left-half of the domain and nonwetting the in right-half of the domain, as shown in Fig. 3.26(b)), the saturations move under capillary forces only. The capillary forces work to homogenize (mix) the saturation in the domain. We assume the domain is heterogeneous and has different sand types in each half. Hence, the two halves have different capillary-pressure models, as shown in Fig. 3.26(a). We simulate the case for 5000 CPVI. The final solution with $\Delta t = 0.1$ (CFL≈2.98) and with $\Delta t = 5000$ (CFL≈47509) are shown in Fig. 3.26(c). The final solutions with both timestep sizes show a saturation jump at the interface of the two half domains due to the heterogeneity of capillary models. Note that $p_c$ is continuous at the interface, but the saturations are discontinuous.

In Fig. 3.27, we show the performance comparison between the standard Newton and the trust-region Newton method. For this capillary mixing case, the standard Newton fails to converge when $\Delta t \geq 0.1$ (CFL≈2.98), whereas our trust-region Newton is able to converge for $\Delta t = 5000$ (CFL≈47509). Fig. 3.27(c) shows that if both Newton-based methods employ their maximum convergent timesteps ($\Delta t = 0.1$ for standard Newton and $\Delta t = 5000$ for trust-region Newton), we achieve around four orders of magnitude reduction in the number of total iterations. This is based on the assumption that the numerical diffusion with the large timestep can be ignored (see Fig. 3.26(c)).
CHAPTER 3. TRUST-REGION NEWTON FOR TRANSPORT

Figure 3.26: Numerical results with trust-region Newton scheme of a 1D incompressible two-phase segregation under (heterogeneous) capillarity ($M^0 = 1$, $N_g = 0$ and $P_e = 0$): (a) capillary models; (b) initial condition; (c) final solution with $\Delta t = 0.1$ (CFL$\approx2.98$) and with $\Delta t = 5000$ (CFL$\approx47508.7$).
Figure 3.27: Performance comparison between standard and trust-region Newton schemes with the number of iterations required to solve a case of 1D incompressible two-phase segregation problem under (heterogeneous) capillarity \((M^0 = 1, N_g = 0\) and \(P_e = 0\)): (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations \((\Delta t = 0.1\) for standard Newton and \(\Delta t = 5000\) for trust-region scheme).
Viscous and gravitational forces

We study 1D transport with both viscous and buoyancy forces. Initially, the left half of the domain ([1,40]) is saturated with the wetting phase and the right half of the domain ([41,80]) is occupied by the nonwetting phase (as shown in Fig. 3.28(a)). We inject the wetting phase at the left boundary (\(S_L = 1.0\)) and produce at constant rate from the right boundary (\(S_R = 0.0\)). The gravity direction is from left to right. Here, \(N_g = -5\), for which the buoyancy forces are more significant. At the end of the simulation, the saturation profiles with \(\Delta t = 0.1\) and \(\Delta t = 5000\) are presented in Fig. 3.28(b); the distribution of fluids approaches the steady-state. Note that while the saturation field appears dispersed when \(\Delta t = 5000\) is used due to large time truncation errors, the solution is consistent with the discrete representation of the physics and is free of oscillations.

![Image](a) initial condition  ![Image](b) final solution

Figure 3.28: Numerical results with trust-region Newton scheme of 1D incompressible two-phase Buckley-leverett displacing problem under viscous and buoyancy forces (\(M^0 = 1, N_g = -5\) and \(P_e = \infty\)): (a) initial condition; (b) final solution with \(\Delta t = 0.1\) (CFL≈1.96) and with \(\Delta t = 5000\) (CFL≈14245).

Fig. 3.29(a) and 3.29(b) show the number of iterations per timestep versus the timestep size and the CFL number. Standard Newton can converge only for timesteps
equal or smaller than 1 CPVI (CFL≈4.69) while the trust-region Newton is convergent even with \( \Delta t = 5000 \) (CFL≈14245). In Fig. 3.29, we compare the number of total iterations between two simulations: standard Newton with \( \Delta t = 1 \) (the maximum convergent timestep for standard Newton) and the trust-region scheme with \( \Delta t = 5000 \). It is observed that we can reduce the total iterations by almost three orders by applying the trust-region solver.

**Gravitational and capillary forces**

For this example, the boundaries are closed, and we study segregation in the presence of both buoyancy and capillary forces. Initially, the wetting (heavier) phase occupies the top-half domain and the nonwetting (lighter) phase is in the bottom-half domain (i.e., same as the initial condition for the case of gravity segregation). The capillary-pressure model is spatially homogeneous. We simulate this case with both the standard and the trust-region Newton schemes for 5000 CPVI. The simulation results with the trust-region scheme are shown in Fig. 3.30.

Fig. 3.30(a) plots the total flux \( F \) defined in Eqn. 3.20 in the presence of buoyancy and capillary forces. For the trust-region scheme, chopping occurs at the inflection points of the total flux, \( F \), and the unit-flux point of flux function, \( f \). The trust-region solver is convergent even when the timestep size is \( \Delta t = 5000 \), i.e., one timestep to go from zero to 5000. Note that there is transition zone between the two phases in the final saturation profile (Fig. 3.30(c)), which is different from the sharp interface in the gravity segregation case.

The statistics of the Newton iterations for both schemes are shown in Fig. 3.31. It is clear that the standard Newton scheme converges for \( \Delta t \in \{0.1, 1\} \), while the trust-region solver can converge for any timestep size. The comparison of the total iterations in Fig. 3.31(c) implies great savings (more than two orders) in the computational cost by applying the trust-region scheme.
Figure 3.29: Performance comparison between the standard and trust-region Newton schemes of 1D two-phase Buckley–Leverett displacing problem under viscous and buoyancy forces ($M^0 = 1$, $N_g = -5$ and $P_e = \infty$): (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations ($\Delta t = 1$ for standard Newton and $\Delta t = 5000$ for trust-region scheme).
Figure 3.30: Numerical results with trust-region Newton scheme of a 1D incompressible two-phase gravity and (homogeneous) capillarity segregation problem with \( M^0 = 1 \). Buoyancy:capillarity=1:1: (a) total flux function (Red dots represent inflection points); (b) initial condition; (c) final solution with \( \Delta t = 0.1 \); (d) final solution with \( \Delta t = 5000 \) (CFL\( \approx 4834 \)).
Figure 3.31: Performance comparison between standard and trust-region Newton schemes with the number of iterations required to solve a case of 1D incompressible two-phase gravity and (homogeneous) capillarity segregation: (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations. $M^0 = 1$, and buoyancy:capillarity=1:1.
Viscous, gravitational and capillary forces

For this example, the simulation is conducted in the presence of viscous, buoyancy and capillary forces. We set $N_g = -5$ and $P_e = 0.2$. The initial condition is $S = 0.8$ in the left-half domain and $S = 0.0$ in the right-half of the domain (Fig. 3.32(b)). The boundary conditions are $S_L = 1.0$ and $S_R = 0.0$, i.e., injecting the wetting phase at the left boundary and producing wetting and nonwetting phases at the right boundary. Note that in this case, the capillary model is heterogeneous, with the two capillary-pressure curves shown in Fig. 3.32(a) used for the left and right half-domains, respectively.

The solutions using the trust-region scheme are shown in Fig. 3.32(c). The solution with $\Delta t$ of 500 CPVI is quite close to that with $\Delta t$ of 0.1 CPVI. Note that there is a kink in the saturation solution, which results from the discontinuity in the capillary model.

Figs. 3.33(a) and 3.33(b) show that the standard Newton scheme is only convergent for small timesteps, i.e., $\Delta t \in \{0.1, 1\}$ or CFL$<10$, while the trust-region scheme converges for any timestep (up to $\Delta t = 500$ or CFL$\approx3491$). We compare the number of total iterations for standard Newton with $\Delta t = 1$, which is its maximum convergent timestep, with the trust-region solver using $\Delta t = 500$ in Fig. 3.33(c). Savings of about two-orders of magnitude are achieved by using the trust-region Newton algorithm.

3.7.4 Comparison with NLEQ-RES algorithm

Up to this point, we have demonstrated the performance of our trust-region Newton method for single-cell, two-cell, and 1D transport problems. In Section 2.4, we described the NLEQ-RES algorithm [30] that is also based on the idea of trust regions. NLEQ-RES, which is not designed specially for multiphase flow in porous media, can be applied to general nonlinear problems. Here, we compare NLEQ-RES and our
Figure 3.32: Numerical results with trust-region Newton scheme of 1D incompressible two-phase Buckley-leverett displacing problem with viscous, gravitational, and heterogeneous capillary fluxes ($M^0 = 1$, $N_g = -5$ and $P_e = 0.2$): (a) capillary models; (b) initial condition; (c) final solution with $\Delta t = 0.1$ and with $\Delta t = 500$ (CFL≈3491).
Figure 3.33: Performance comparison between standard and trust-region Newton schemes with the number of iterations required to solve a case of 1D incompressible two-phase Buckley-Leverett displacing problem with viscous, gravitational, and heterogeneous capillary fluxes ($M_0 = 1$, $N_g = -5$ and $P_e = 0.2$): (a) iterations vs. timestep; (b) iterations vs. CFL; (c) total iterations ($\Delta t = 1$ for standard Newton and $\Delta t = 500$ for trust-region scheme).
trust-region Newton method (i.e., number of Newton iterations) of the two nonlinear solvers. The comparisons are conducted for two-cell and 1D transport problems. We restrict the comparison to the "pure" transport problems (i.e., the total-velocity is fixed, and the pressure distribution is not coupled to the nonlinear scalar transport-saturation-equation). This is because for multi-dimensional problems with coupled flow and transport, which is the target of this work, we have found it near-impossible to "tune" the large number of parameters needed by these "general" nonlinear solver packages. These difficulties were encountered even for cases where the objective was to find the "right" parameters for a single simulation run, let alone finding a set of parameters that works well for wide variation of the flow and transport physics within this class of nonlinear immiscible porous-media problems. The flexibility and generality of NLEQ-RES and similar general (black-box) packages have proved to be quite difficult to reconcile with the need for nonlinear solution strategies that capture the specific nonlinearities associated with multiphase flow in natural porous media.

Two-cell transport

First, we study a numerical example of two-cell transport problem with only viscous forces. Initially, the two cells are fully saturated by nonwetting phase, and the wetting phase is injected at the left boundary. The final solution at the end of simulation is $[S_1, S_2] = [0.7753, 0.7057]$. We plot the contours of residual norm (thin color lines), Newton flow (thick black lines), and the iterative solutions (thick red dots) for both the NLEQ-RES and our trust-region Newton method, respectively, in Fig. 3.34. For definition of Newton flow, see Section 3.7.2. Newton flows are always along the descent direction of residual norms $33$. In Fig. 3.34(a), it is observed that the iterative solutions from the NLEQ-RES, ‘loosely follow’ one Newton flow, which is starting from the initial condition. On the contrary, Fig. 3.34(b) shows that iterative solutions from our trust-region Newton method cross different Newton flows and
converge to the solution quickly, with monotonic decreasing in residual norms from one iteration to the next. For this numerical example, NLEQ-RES converges within 12 iterations while our trust-region Newton converges within 6 iterations.

![Newton flow for two-cell transport with viscous flux.](image)

Figure 3.34: Newton flow for two-cell transport with viscous flux. \( N_g = 0, \ P_e = \infty, \ M^0 = 1, \ \Delta t = 10, \ S_L = 1.0, \) and \( S_R = 0.0. \) The initial guess is \( S_1 = 0.0 \) and \( S_2 = 0.0. \) \([S_1^{n+1}, S_2^{n+1}]=[0.7753, 0.7057].\) (a) NLEQ-RES; (b) trust-region Newton.

Now we compare the performance between the NLEQ-RES and our trust-region Newton method using a two-cell transport problem with both viscous and buoyancy forces. Starting from nonwetting saturated condition, we inject wetting phase at the left boundary. Let \( N_g = -5 \) and buoyancy direction is the same as the total-velocity direction. The final solution is \([S_1, S_2]=[0.6104, 0.6907].\) We plot the iterative solutions for both the NLEQ-RES and our trust-region Newton method in Fig. 3.35. The NLEQ-RES converges to the solution within 10 iterations while our trust-region Newton takes 8 iterations. In Fig. 3.35 there are kinks in the Newton flows, due to the discontinuity introduced by upwinding scheme and flow reversal. The iterative solutions from the NLEQ-RES (Fig. 3.35(a)) follow the Newton flow starting from the initial condition to guarantee that the residual norm in each iteration is smaller than that in the previous iteration. On the other hand, the iterative solutions from our trust-region Newton method (Fig. 3.35(b)) cross different Newton flows.
CHAPTER 3. TRUST-REGION NEWTON FOR TRANSPORT

Figure 3.35: Newton flow for two-cell transport with viscous and gravitational flux. \(N_g = -5, \ P_e = \infty, \ M^0 = 1, \ \Delta t = 10, \ S_L = 0.6, \) and \(S_R = 0.8. \) The initial guess is \(S_1 = 0.0 \) and \(S_2 = 0.0. \) \([S_1^{n+1}, S_2^{n+1}] = [0.6104, 0.6907].\) (a) NLEQ-RES; (b) trust-region Newton.

1D transport

Here, we investigate 1D (80 grid blocks) transport problems. First, we study a 1D transport problem with only viscous forces, then we move to a gravity segregation problem in 1D domain.

For the 1D transport problem with only viscous forces, initially, the 1D domain is fully saturated by nonwetting phase and we inject wetting phase at the left boundary. The total simulation time is \(T = 100 \) CPVI, and we simulate the transport problem using two different timestep sizes, respectively, \(\Delta t = 10 \) (CFL(max)=20) and \(\Delta t = 100\) (CFL(max)=200). For these two timestep sizes, the number of Newton iterations per timestep is summarized in Table 3.3. It shows that for both timestep sizes, our trust-region Newton method converges faster than the NLEQ-RES. Also, for the larger timestep size (i.e., \(\Delta t = 100\)), the advantage of trust-region Newton over the NLEQ-RES is more significant.

Then, we study a segregation problem in the 1D domain. Initially wetting (heavier) phase resides in the top-half domain and nonwetting (lighter) phase resides...
Table 3.3: Performance comparison between the NLEQ-RES and trust region Newton scheme for 1D transport problem with viscous forces \((N_g = 0, P_e = \infty)\) with \(M^0 = 1\).

<table>
<thead>
<tr>
<th>total time</th>
<th>(\Delta t)</th>
<th>CFL (max)</th>
<th>NLEQ-RES iter/ts</th>
<th>trust region iter/ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>20</td>
<td>8.2</td>
<td>6.8</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>200</td>
<td>19</td>
<td>9</td>
</tr>
</tbody>
</table>

in the bottom-half domain (as shown in Fig. 3.36(a)). The total simulation time is \(T = 1000\) CPVI, and we simulate the transport problem using three timestep sizes: \(\Delta t = 10\) (CFL(max) = 5.96), \(\Delta t = 100\) (CFL(max) = 44.68), and \(\Delta t = 1000\) (CFL(max) = 328.75). Fig. 3.36(b) shows that there is time truncation error associated with large timestep sizes. For the three timestep sizes, we summarized the

Figure 3.36: Numerical results with a 1D incompressible two-phase gravity segregation \((M^0 = 1, N_g = -\infty\) and \(P_e = \infty)\): (a) initial condition; (b) final solution with \(\Delta t = 10\) (CFL\(\approx\)5.96) and with \(\Delta t = 1000\) (CFL\(\approx\)328.75).

number of Newton iterations per timestep for both our trust-region Newton method and the NLEQ-RES in Table 3.4. It shows that for all three timestep sizes, our trust-region Newton method takes fewer iterations to converge than the NLEQ-RES.
When timestep is small (i.e., $\Delta t = 10$), the numbers of Newton iterations for both nonlinear solvers are quite close (3.19 vs. 3.15). When timestep is relatively large, i.e., $\Delta t = 100$, the difference in the number of iterations between the two nonlinear solvers is quite significant, i.e., 178.5 vs. 6.6. When a huge timestep ($\Delta t = 1000$) is used, the NLEQ-RES is not convergent while our trust-region Newton takes only 20 iterations to converge. Thus, our trust-region Newton method yields much better nonlinear performance than the NLEQ-RES.

Table 3.4: Performance comparison between NLEQ-RES and trust region Newton scheme for 1D gravity segregation ($N_g = -\infty$, $P_e = \infty$) with $M^0 = 1$.

<table>
<thead>
<tr>
<th>total time</th>
<th>$\Delta t$</th>
<th>CFL (max)</th>
<th>NLEQ-RES iter/ts</th>
<th>trust region iter/ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>10</td>
<td>5.96</td>
<td>3.19</td>
<td>3.15</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>44.68</td>
<td>178.5</td>
<td>6.6</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>328.75</td>
<td>-</td>
<td>20</td>
</tr>
</tbody>
</table>

Discussion on Performance Comparison

From the performance comparison in two-cell and 1D problems, we can see that our trust-region Newton method has superior nonlinear performance compared with NLEQ-RES, which is one of the popular general trust-region software libraries. Our method defines the trust regions before the Newton iterations are started and converges to the solution very reliably and efficiently. First, from the comparison of nonlinear performance, we can see that for all the numerical cases, our trust-region Newton method takes fewer iterations to converge than the NLEQ-RES algorithm. Moreover, for each Newton iteration, the computational cost between the two nonlinear solvers are different. There is an inner loop within each Newton iteration in NLEQ-RES, where the damping factor is calculated with a try-adapt-try strategy (i.e., predict the damping factor and then correct it according to the residual norm
with the predicted damping factor). Also, by using the NLEQ-RES, a set of monitoring quantities are computed to guide the iterative solutions to ‘loosely’ follow the Newton flow. Hence, the computational cost per Newton iteration by using the NLEQ-RES is higher than that by using our trust-region Newton method. Finally, heuristic parameters are used NLEQ-RES. These parameters case-dependent. For each new nonlinear problem, we need to tune these parameters to achieve optimal performance with NLEQ-RES. On the contrary, for our trust-region Newton method, there are no case-dependent parameters.

3.7.5 Large heterogeneous examples

The trust-region Newton solver is designed to overcome the convergence difficulties associated with strongly nonlinear transport problems. In the following examples, we show how the trust-region solver improves the convergence behaviors for coupled flow and transport problems in highly heterogeneous 3D reservoir models.

We implemented the trust-region solver in AD-GPRS (Automatic Differentiation General Purpose Research Simulator) [17, 28, 51], which is an in-house general-purpose research simulator developed in SUPRI-B (reservoir simulation research group at Stanford university). To demonstrate the effectiveness of our new nonlinear solver for large-scale, complex heterogeneous models, we studied two-phase displacements using the SPE 10 model [52] \((60 \times 220 \times 85 = 1.12 \text{ million grid blocks})\). The top 35 layers of the reservoir represent a Tarbert formation with highly variable permeabilities ranging from \(7.1 \times 10^{-4}\) to \(2.0 \times 10^{4}\) md. The bottom 50 layers represent an UpperNess sequence, which is highly channelized and has a wide permeability range of \(6.6 \times 10^{-4}\) to \(2.0 \times 10^{4}\) md. The relative permeability curves for this two-phase displacement (piecewise linear curves from the SPE 9 model) is displayed in Fig. 3.37. The viscosity ratio is 10.
CHAPTER 3. TRUST-REGION NEWTON FOR TRANSPORT

First, we study the behavior of a water flood (i.e., water displacing oil) problem in the upper zone of SPE 10 model (Tarbert formation) \((60 \times 220 \times 35 = 462,000\) grid blocks). Water is injected at the center of the reservoir at \(87.5\ m^3/\text{day} (8 \times 10^{-5}\ \text{pore volumes per day})\) through an injector that fully perforates the reservoir; four production wells, which are also fully perforates the reservoir, are located in the four corners of the reservoir, operating at a bottom hole pressure of 4000 psi. All the other reservoir and fluid properties are the same as the public data in [52]. The simulations are carried out up to \(T = 10\) days, which corresponds to \(8 \times 10^{-4}\) pore volumes injected (PVI). For any timestep, if the nonlinear solver does not converge within 30 iterations, the iterations are stopped, and the current timestep is cut in half before restarting. We performed several simulations, each with a fixed constant target timestep, namely, 0.1, 1 and 10 days. Buoyancy cannot be ignored and counter-current flow occurs in \(7\% \sim 13\%\) of all the vertical (control-volume) interfaces for a typical timestep during the simulation, for all the target timestep sizes (0.1 day, 1 day, and 10 day). By inspecting the CFL number distribution over all the reservoir cells in Fig. 3.38 for timestep size as 10 days, we find that the CFL number varies by orders of magnitude in the reservoir. Even when the model is simulated for up to \(8 \times 10^{-4}\) PVI, i.e., only

![Relative permeability curves in large test case.](image)

Figure 3.37: Relative permeability curves in large test case.

**Tarbert formation**

First, we study the behavior of a water flood (i.e., water displacing oil) problem in the upper zone of SPE 10 model (Tarbert formation) \((60 \times 220 \times 35 = 462,000\) grid blocks). Water is injected at the center of the reservoir at \(87.5\ m^3/\text{day} (8 \times 10^{-5}\ \text{pore volumes per day})\) through an injector that fully perforates the reservoir; four production wells, which are also fully perforates the reservoir, are located in the four corners of the reservoir, operating at a bottom hole pressure of 4000 psi. All the other reservoir and fluid properties are the same as the public data in [52]. The simulations are carried out up to \(T = 10\) days, which corresponds to \(8 \times 10^{-4}\) pore volumes injected (PVI). For any timestep, if the nonlinear solver does not converge within 30 iterations, the iterations are stopped, and the current timestep is cut in half before restarting. We performed several simulations, each with a fixed constant target timestep, namely, 0.1, 1 and 10 days. Buoyancy cannot be ignored and counter-current flow occurs in \(7\% \sim 13\%\) of all the vertical (control-volume) interfaces for a typical timestep during the simulation, for all the target timestep sizes (0.1 day, 1 day, and 10 day). By inspecting the CFL number distribution over all the reservoir cells in Fig. 3.38 for timestep size as 10 days, we find that the CFL number varies by orders of magnitude in the reservoir. Even when the model is simulated for up to \(8 \times 10^{-4}\) PVI, i.e., only
a small portion of the reservoir is flooded, the maximum CFL number is beyond 500 and almost 1% (~46,200) grid blocks have CFL numbers larger than unity, which indicates potential convergence challenges for the existing nonlinear solvers.

Figure 3.38: CFL distribution for upper zone of SPE 10 model (Tarbert formation, \(60 \times 220 \times 35 = 462,000\)): two-phase flow with gravity. End of simulation: \(T = 10\) days. Timestep sizes: \(\Delta t = 10\).

Fig. 3.39, 3.40 and Table 3.5 show the convergence performance (total number of iterations and total simulation time) for three nonlinear solvers: standard Newton, Newton with Appleyard chopping \([49, 32]\), and trust-region Newton.

Table 3.5: Summary of performance for upper zone of SPE 10 model (Tarbert formation, \(60 \times 220 \times 35 = 462,000\)): two-phase flow with gravity. End of simulation: \(T = 10\) days. ‘Wasted timesteps’ and ‘wasted iterations’ indicate the number of timesteps and Newton iterations that are wasted due to timestep cuts.

<table>
<thead>
<tr>
<th></th>
<th>standard (\Delta t = 0.1)</th>
<th>standard (\Delta t = 1)</th>
<th>standard (\Delta t = 10)</th>
<th>Appleyard (\Delta t = 0.1)</th>
<th>Appleyard (\Delta t = 1)</th>
<th>Appleyard (\Delta t = 10)</th>
<th>trust region (\Delta t = 0.1)</th>
<th>trust region (\Delta t = 1)</th>
<th>trust region (\Delta t = 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>timesteps</td>
<td>107</td>
<td>62</td>
<td>58</td>
<td>100</td>
<td>22</td>
<td>22</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>iterations</td>
<td>328</td>
<td>239</td>
<td>233</td>
<td>359</td>
<td>420</td>
<td>434</td>
<td>422</td>
<td>88</td>
<td>23</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>15</td>
<td>174</td>
<td>291</td>
<td>0</td>
<td>28</td>
<td>79</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>148</td>
<td>2829</td>
<td>3847</td>
<td>0</td>
<td>639</td>
<td>2137</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>total iterations</td>
<td>476</td>
<td>3068</td>
<td>4080</td>
<td>359</td>
<td>1059</td>
<td>2571</td>
<td>422</td>
<td>88</td>
<td>23</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>5190</td>
<td>30318</td>
<td>40152</td>
<td>3136</td>
<td>8364</td>
<td>21091</td>
<td>10351</td>
<td>2078</td>
<td>615</td>
</tr>
</tbody>
</table>

In Fig. 3.39 we observe that when the timestep size is small (\(\Delta t = 0.1\)), all the
Figure 3.39: Comparison of total Newton iterations for upper zone of SPE 10 model (Tarbert formation, $60 \times 220 \times 35 = 462,000$): two-phase flow with gravity. End of simulation: $T = 10$ days. Different target timestep sizes are tested: $\Delta t = 0.1$, $\Delta t = 1$, and $\Delta t = 10$, respectively.

Figure 3.40: Comparison of total simulation time for upper zone of SPE 10 model (Tarbert formation, $60 \times 220 \times 35 = 462,000$): two-phase flow with gravity. End of simulation: $T = 10$ days. Timestep size: $\Delta t = 10$. 
three nonlinear solvers converge for all the steps (except that the standard Newton employs a few timestep cuts), and that they consume around the same number of total Newton iterations. As the timestep size increases ($\Delta t = 1, 10$), the trust-region solver can converge without difficulties, whereas both the standard Newton and Newton with Appleyard chopping suffer from convergence difficulties and need to cut the timestep multiple times in order to achieve convergence. Timestep cuts result in a significant number of wasted nonlinear iterations and a serious degradation in the overall computational performance. Taking too large a timestep in standard Newton or Newton with Appleyard chopping actually makes the simulation slower, due to the prohibitively large numbers of timestep cuts and wasted iterations, whereas the opposite is true for trust-region Newton method. Indeed, trust-region Newton method with $\Delta t = 10$ takes only 0.5% of the total Newton iterations needed by standard Newton (23 vs. 4080). An initial (unoptimized) implementation for the trust-region solver achieves a speed-up of 65 times compared with the standard Newton method for $\Delta t = 10$ (shown in Fig. 3.40). While increasing the timestep size increases the time truncation error, our objective is to first remove limits on the timestep due to the nonlinear solver itself. Once convergence of the nonlinear solver is not the limiting factor, we can choose the timestep size based on accuracy consideration. Table 3.5 summarizes timesteps, total iterations and simulation time for all three nonlinear solvers, respectively, with $\Delta t = 0.1, 1, \text{and } 10$.

Now we compare the Newton iterations per converged timestep for standard Newton, Newton with Appleyard chopping, and trust-region Newton in Fig. 3.41. For standard Newton, since a large number of timesteps (i.e., 62) is required to simulate this water-flood case up to $t = 10$, the number of iterations per converged step is relative small (no more than 5 iterations per timestep). Compared with trust-region Newton, it is observed that Newton with Appleyard chopping takes more steps until $t = 10$ (i.e., 22 vs. 10) and much more iterations per step. The trust-region Newton
Figure 3.41: Comparison of Newton iterations per converged timestep for upper zone of SPE 10 model (Tarbert formation, $60 \times 220 \times 35 = 462,000$) with target timestep size $\Delta t = 1$: two-phase flow with gravity. End of simulation: $T = 10$ days.

The full SPE 10 model

We now discuss the results for the same water flood problem but using the full SPE 10 model ($60 \times 220 \times 85 = 1.12$ million grid blocks) with water injection rate at 212.5 m$^3$/day ($9.8 \times 10^{-5}$ PVI/day). End of simulation is $T = 10$. We compare the convergence performance for Newton with Appleyard chopping with our trust-region algorithm for $\Delta t = 0.1$, $\Delta t = 1$ and $\Delta t = 10$ (Here, we did not display the convergence performance of the standard Newton for the full SPE 10 model since its performance is much inferior to Appleyard and trust-region Newton methods). As shown in Fig. 3.42, for all the tested timestep sizes, the trust-region Newton method takes less Newton iterations than Newton with Appleyard chopping. This advantage become significant as the timestep size increases. Note that for $\Delta t = 10$, trust-region Newton method takes only 0.5% (74 vs. 4902) of the total Newton iterations taken by the Newton method with Appleyard chopping. The resulting saving of computational
effort (i.e., total simulation time) is illustrated in Fig. 3.43. For $\Delta t = 10$, compared with Newton method with Appleyard chopping, a speed-up of around 20 (109944 vs. 5861) is achieved by using the trust-region Newton method. The timesteps, total iterations and total simulation time for both nonlinear solvers are summarized in Table 3.6 respectively, for $\Delta t = 0.1$, $\Delta t = 1$ and $\Delta t = 10$. 

We also studied the long-time simulation performance (up to 2040 days, 0.2PVI) of the trust-region Newton method for the full SPE 10 model. The simulation starts with
Table 3.6: Summary of performance for SPE 10 model (60 × 220 × 85 = 1.12 million): two-phase flow with gravity. End of simulation: $T = 10$ days. ‘Wasted timesteps’ and ‘wasted iterations’ indicate the number of timesteps and Newton iterations that are wasted due to timestep cuts.

<table>
<thead>
<tr>
<th></th>
<th>Appleyard</th>
<th></th>
<th></th>
<th>trust region</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\Delta t = 0.1$</td>
<td>$\Delta t = 1$</td>
<td>$\Delta t = 10$</td>
<td></td>
<td>$\Delta t = 0.1$</td>
<td>$\Delta t = 1$</td>
</tr>
<tr>
<td>timesteps</td>
<td>102</td>
<td>42</td>
<td>35</td>
<td>102</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>iterations</td>
<td>622</td>
<td>776</td>
<td>676</td>
<td>455</td>
<td>83</td>
<td>43</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>3</td>
<td>95</td>
<td>150</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>38</td>
<td>2272</td>
<td>4226</td>
<td>18</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>total iterations</td>
<td>660</td>
<td>3048</td>
<td>4902</td>
<td>473</td>
<td>83</td>
<td>74</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>12879</td>
<td>77156</td>
<td>109944</td>
<td>28728</td>
<td>5044</td>
<td>5861</td>
</tr>
</tbody>
</table>

a timestep size of 5 days, and doubles the timestep size when the nonlinear solution converges and cuts the timestep size by half when the nonlinear solution fails. The maximum allowable timestep size is 320 days. For each time step, counter-current flow occurs at 5% to 10% of the vertical cell interfaces. Fig. 3.44 shows the maximum CFL number for each time step during the simulation. We observe that as the simulation time increases, the CFL number increases rapidly with the growing timestep size and reaches its maximum value of 81935.4 at the largest timestep size ($\Delta t = 320$ days). The maximum CFL number averaged for all time steps is 15588.8. Compared with the previous examples (short-term simulation, $9.8 \times 10^{-4}$ PVI), the CFL numbers in this problem are extremely large, whereas the trust-region Newton method can still converge in most of the cases. That is, the trust-region Newton method can provide superior nonlinear performance even for large timestep sizes (corresponding to huge CFL numbers) during a long-term simulation.
Figure 3.44: History of the maximum CFL number for full SPE 10 model ($60 \times 220 \times 85 = 1.12$ million) simulated using trust-region nonlinear solver with a minimum timestep size of 5 days and a maximum timestep size of 500 days. End of simulation: $T = 2040$ days (0.2PVI).
Chapter 4

Trust-Region Newton Solver for Coupled Flow and Transport

For incompressible multiphase flow and transport in one dimension (1D), if the total-velocity is constant, then the saturation distribution can be obtained by solving the transport problem only. The convergence difficulties associated with the nonlinear transport problem were addressed in Chapter 3. The trust-region Newton solver described there allows for using arbitrary timestep sizes in 1D, such that one can choose the timestep based on accuracy requirements, as opposed to worrying about the ability of the nonlinear solver to converge. In 2D or 3D domains, the assumption that the total-velocity does not vary in time and space is no longer valid, and we need to solve the coupled flow (pressures and velocities of the fluid phases) and transport (saturation) problems for each timestep. On one hand, the coupling is due to the strong dependence of the saturation field on the fractional flow and total velocity, which is calculated from the pressure distribution. On the other hand, changes in the saturation field alter the phase mobilities, which in turn affect the pressure distribution. Buoyancy adds to the complexity by introducing phase-flow reversal as a function of time (or Newton iteration), which will be discussed in Section
As described in Section 1.1, the coupled system usually exhibits a mixed near elliptic-hyperbolic character [1, 14]. Many numerical formulations take the mixed character of the equations into consideration, whereby the coupled system is separated into elliptic (pressure) and hyperbolic (saturation) parts [15, 5, 6].

The sequential-implicit method (SIM) is one of the numerical formulations that splits the overall problem into two parts - flow and transport - and deals with each part separately. For the flow problem, the mass conservation equations are combined and expressed in terms of pressure (Eqn. 1.7) and are then solved implicitly. Then, the total-velocity is calculated using the computed pressure field. The total-velocity is then fixed in the transport problem (Eqn. 1.12), which is solved implicitly for the saturation field. The sequential-implicit method (SIM) has been employed in the framework of multiscale methods, such as MSFV [22, 26]. These methods rely on sequential solutions of the flow and transport problems. Extensive numerical experience indicates that SIM works well for viscous-dominated problems. However, SIM suffers from convergence difficulties in the presence of significant buoyancy when the timestep sizes are relatively large.

When buoyancy forces are significant, the flow directions of the two phases (at an interface between two cells) based on the pressure solution can change as a function of iteration and time. When the timestep size is relatively large, the phase-flow directions obtained from the pressure solution may change when they get updated based on the saturation field computed from the transport problem. The updated phase-flow directions can change again when they are updated based on the pressure solution computed in the next outer iteration of the sequential scheme. Changes in the flow direction of the fluid phases (i.e., flow reversal) between sequential updates of the pressure and saturation fields can slow down outer-loop convergence quite significantly. In cases where large timesteps (CFL $\gg 1$) are taken, SIM may not converge at all, thus requiring timestep cuts. It is observed numerically in [27] that
when MSFV is embedded into SIM, extremely small timesteps (CFL ≪ 1) have to be used in the presence of buoyancy. In order to improve the convergence behavior of sequential-implicit methods for coupled flow and transport, we propose a nonlinear solution strategy that deals with flow reversal effectively.

In this chapter, we first investigate the convergence limitation (i.e., flow reversal) associated with the sequential-implicit method when buoyancy forces are significant (Eqn. 1.7 and 1.12). Then, we present a nonlinear solver for coupled multiphase flow and transport that converges for extremely large time steps.

### 4.1 Flow reversal

Flow reversal refers to changes in the directions of the fluid phases at cell interfaces as a function of iteration, or time. It is observed that if no flow reversal occurs in the transport problem, the sequential-implicit method (SIM) is always convergent. On the other hand, if flow reversal occurs during transport, then the pressure field will not be consistent with the updated phase mobilities and flow directions. And even if the saturation can converge unconditionally using our trust-region Newton method in Chapter 3, the phase-flow directions may keep changing from one iteration (outer loop) to the next. In such cases, SIM will not converge.

The occurrence of flow reversal is sensitive to the timestep size. For a given initial condition, when a relatively small timestep size is used, we may not observe flow reversal, even in the presence of strong buoyancy forces, and the sequential-implicit method will converge. Flow reversal may occur, however, for the same initial condition when the timestep size is slightly larger, resulting in flow reversal and non-convergence of SIM. There is no a-priori way to determine the limiting timestep size beyond which the sequential-implicit method will fail to converge.

Here, we investigate flow reversal as a function of time or iteration. Assuming
the total-velocity field to be fixed, we will show that the sign of the total-velocity at an interface determines the occurrence of flow reversal and identifies the specific phase (nonwetting, or wetting) that switches its flow direction as a function of time, or Newton iteration.

Recall that $N_g$ represents the ratio of buoyancy to viscous forces, namely,

$$N_g = \frac{k(\rho_w - \rho_n) g \nabla h}{\mu_n u_t}.$$  \hspace{1cm} (4.1)

Since the upward $z$ direction is assumed positive and we assume that $h$ (height) is positive upwards, we have $\nabla h = 1$. Without loss of generality, we assume $\rho_w > \rho_n$. With $\nabla h = 1$ and with $\rho_w > \rho_n$, Eqn. (4.1) indicates that $N_g$ is positive when the total-velocity ($u_t$) is positive (i.e., against gravity). We plot the flux functions with $N_g = -5$ and $N_g = 5$ in Fig. 4.1 as examples (recall that $S = S_w$ and $f = f_w$ denote the wetting-phase saturation and flux, respectively). Note that there is a unit-flux point (orange dot), where the flux function is unity, for $N_g < 0$ ($u_t < 0$). There is a zero-flux point (green dot), where the flux function equals to zero, for $N_g > 0$ ($u_t > 0$). For $N_g < 0$, when the saturation crosses the unit-flux point, $S^{f=1}$, the nonwetting-phase flow direction flips; for $N_g > 0$, the zero-flux point $S^{f=0}$ is the saturation at which the wetting-phase flow direction flips.

In Fig. 4.1, the arrows with labels (a), (b), (c), and (d) correspond to the four cases of flow reversal. Figs. 4.2(a) - 4.2(d) describe the details of these cases. Fig. 4.2(a) and 4.2(b) show switching from cocurrent to counter-current flow, and Fig. 4.2(c) and 4.2(d) show switching from counter-current to cocurrent flow.

In Fig. 4.2(a), initially both the wetting and nonwetting phases are moving from cell $i + 1$ to cell $i$ (cocurrent flow), i.e., the same (negative) direction as the total-velocity and the gravity direction ($N_g < 0$). Once the wetting-phase saturation in cell $i + 1$ increases to $S^{f=1}$, the unit-flux point in Fig. 4.1 the nonwetting phase flips its
Figure 4.1: Flux functions for two-phase flow in the presence of strong buoyancy, respectively, with \( N_g = -5 \) (blue line) and \( N_g = 5 \) (red line). Orange dot represents unit-flux point \( (S_f = 1) \) while green dot is zero-flux point \( (S_f = 0) \). The arrows correspond to the four types of flow reversal in Fig. 4.2.

flow direction from flowing out of cell \( i + 1 \) to flowing into it. Note that the wetting phase continues to flow out of cell \( i + 1 \), i.e., the same (negative) direction as \( u_t \). Thus, flow reversal of the nonwetting-phase is caused by an increase in the wetting-phase saturation of cell \( i + 1 \), and is not caused by a change in the saturation of cell \( i \). That is, only a saturation change in the upwind cell w.r.t.(with respect to) \( u_t \) (cell \( i + 1 \) in this case) matters.

In Fig. 4.2(b), initially \( u_w, u_n \) and \( u_t \) are all flowing from cell \( i \) to cell \( i + 1 \), (i.e., opposite to the gravity direction \( (N_g > 0, u_t > 0) \)). When \( N_g > 0 \), flow reversal occurs only in the wetting phase. Once the wetting-phase saturation of cell \( i \) decreases to \( S_f = 0 \), the zero-flux point in Fig. 4.1, the wetting-phase flow direction flips, and counter-current flow occurs. This is induced by the decrease of the wetting phase in cell \( i \) only and is independent of the saturation change in cell \( i + 1 \). To summarize, flow reversal from cocurrent to counter-current flow at the cell interface depends only
Figure 4.2: Four types of flow reversal under strong buoyancy: (a) $N_g < 0$, cocurrent flow switching to counter-current flow; (b) $N_g > 0$, cocurrent flow switching to counter-current flow; (c) $N_g < 0$, counter-current flow switching to cocurrent flow; (d) $N_g > 0$, counter-current flow switching to cocurrent flow.
on the saturation change in the upwind cell w.r.t. the total velocity.

In Fig. 4.2(c), the flow flips from counter-current to cocurrent, due to a decrease in $S_{i+1}$. Since $N_g < 0$, flow reversal occurs in the nonwetting phase. Once $S_{i+1}$ decreases and crosses $S^f=1$ (as the label in Fig. 4.1 shows), the nonwetting-phase flow direction at the interface flips from flowing out of cell $i$ to flowing into cell $i$, while the wetting-phase flow direction remains the same. Therefore, the flow regime becomes cocurrent, i.e., both phases flow in the same direction as the total velocity.

In Fig. 4.2(d), the flow flips from counter-current to cocurrent as in Fig. 4.2(c), but due to an increase in $S_i$. Since $N_g > 0 (u_t > 0)$, flow reversal occurs in the wetting phase. When the wetting-phase saturation in cell $i$ increases up to $S^f=0$ on the flux function, the wetting-phase flow direction switches to the same direction as the nonwetting-phase and the total velocity. For the type of flow reversal depicted by Fig. 4.2(d), the flip in the phase-flow direction is independent of the saturation change in cell $i + 1$.

Note that the upwind cell w.r.t. $u_t$ is: $i + 1$ when $N_g < 0$, and $i$ when $N_g > 0$. It is observed that for any case in Fig. 4.2, flow reversal occurs once the saturation of the upwind cell w.r.t. $u_t$ crosses $S^f=1 \ (N_g < 0)$, or $S^f=0 \ (N_g > 0)$. We define $S^f=1 \ (for \ N_g < 0)$ and $S^f=0 \ (for \ N_g > 0)$ as the ‘flipping points’ that pinpoint flow reversal.

**Proposition 4.1.1.** In the transport iterations, assuming that the total-velocity field is fixed, flipping of the phase-flow direction at a cell interface is always caused by the saturation at the upwind cell w.r.t. the total-velocity crossing $S^f=1$, or $S^f=0$. Assume that flow is positive when it is from cell $i$ to cell $i + 1$, and gravity direction is from cell $i + 1$ to cell $i$. Then, for $N_g > 0 (u_t > 0)$

$$
\begin{align*}
\lambda_{w}^{i+\frac{1}{2}} &= \lambda_w(S_i) \ and \ \lambda_{n}^{i+\frac{1}{2}} = \lambda_n(S_i), \quad S^f=0 < S_i \leq 1 \\
\lambda_{w}^{i+\frac{1}{2}} &= \lambda_w(S_{i+1}) \ and \ \lambda_{n}^{i+\frac{1}{2}} = \lambda_n(S_i), \quad 0 \leq S_i \leq S^f=0
\end{align*}
$$

(4.2)
and for \( N_g < 0 \) (\( u_t < 0 \))

\[
\begin{align*}
\lambda^{i+\frac{1}{2}}_w &= \lambda_w(S_{i+1}) \quad \text{and} \quad \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_i), \quad S^{i+1}_f < S_{i+1} \leq 1 \\
\lambda^{i+\frac{1}{2}}_w &= \lambda_w(S_{i+1}) \quad \text{and} \quad \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_{i+1}), \quad 0 \leq S_{i+1} \leq S^{i+1}_f.
\end{align*}
\] (4.3)

**Proof.** To prove this proposition, we focus on the interface between cells \( i \) and \( i + 1 \). We use \( i + 1/2 \) to denote the interface. If \( u_t > 0 \), the upwind cell w.r.t. \( u_t \) is cell \( i \), while if \( u_t < 0 \), the upwind cell w.r.t. \( u_t \) is cell \( i + 1 \).

The phase mobilities, \( \lambda^{i+1/2}_\alpha \) \((\alpha = w, n)\), at the interface are evaluated using the upstream saturation w.r.t. the phase-flow direction:

\[
\lambda^{i+\frac{1}{2}}_\alpha = \left\{ \begin{array}{ll}
\lambda_\alpha(S_i), & \frac{p_i - p_{i+1}}{\Delta x} + \rho_\alpha g > 0, \\
\lambda_\alpha(S_{i+1}), & \text{otherwise}.
\end{array} \right.
\] (4.4)

In light of Eqn. 1.11, we can rewrite the upstream conditions as

\[
\begin{align*}
\lambda^{i+\frac{1}{2}}_w &= \left\{ \begin{array}{ll}
\lambda_w(S_i), & u_t - kg \Delta \rho \lambda^{i+\frac{1}{2}}_n > 0 \\
\lambda_w(S_{i+1}), & \text{otherwise}
\end{array} \right., \\
\lambda^{i+\frac{1}{2}}_n &= \left\{ \begin{array}{ll}
\lambda_n(S_i), & u_t + kg \Delta \rho \lambda^{i+\frac{1}{2}}_w > 0 \\
\lambda_n(S_{i+1}), & \text{otherwise}
\end{array} \right.. \quad (4.5)
\]

In Eqn. 4.5, the pressure dependence has been eliminated. Nevertheless, Eqn. 4.5 still does not explicitly define the upstream direction of \( \lambda_\alpha \) at the interface \( i + 1/2 \). This is because the latter is defined in terms of the (yet undetermined) mobility of the other phase at the interface. Brenier and Jaffré [45] showed how to explicitly determine the
upstream direction for a given saturation field. Namely,

\[
\lambda^{i+\frac{1}{2}}_w = \begin{cases} 
\lambda_w(S_i), & \theta_w > 0 \\
\lambda_w(S_{i+1}), & \text{otherwise}
\end{cases},
\]

\[
\lambda^{i+\frac{1}{2}}_n = \begin{cases} 
\lambda_n(S_i), & \theta_n > 0 \\
\lambda_n(S_{i+1}), & \text{otherwise}
\end{cases},
\]

where

\[
\theta_w = u - kg\Delta\rho\lambda_n(S_i),
\]

\[
\theta_n = u + kg\Delta\rho\lambda_w(S_{i+1}).
\]

They proved that the upwinding scheme in Eqn. 4.5 is equivalent to that of Eqn. 4.6. Thus, for phase-based upwinding, the phase-flow directions are defined in terms of cell-centered saturation values in either cell \(i\) or \(i+1\). Next, we discuss the upwinding dependence for two cases, \(N_g > 0\) and \(N_g < 0\).

On the one hand, when \(N_g < 0\) (downdip), \(u < 0\). From Eqn. 4.7 it is obvious that \(\theta_w\) is always negative. So we obtain

\[
\begin{cases} 
\lambda^{i+\frac{1}{2}}_w = \lambda_w(S_{i+1}) \text{ and } \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_i), & \theta_w \leq 0 \leq \theta_n \\
\lambda^{i+\frac{1}{2}}_w = \lambda_w(S_{i+1}) \text{ and } \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_{i+1}), & \theta_w \leq \theta_n \leq 0
\end{cases}
\]

By the monotonicity of \(\lambda_w\), there exists a unique \(0 < S_{i+1}^f \leq 1\), such that \(u = -kg\Delta\rho\lambda_w(S_{i+1})\), and thus \(\theta_n = 0\). Eqn. 4.8 is elaborated further by Kwok and Tchelepi in 44 into a more explicit form:

\[
\begin{cases} 
\lambda^{i+\frac{1}{2}}_w = \lambda_w(S_{i+1}) \text{ and } \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_i), & S_{i+1}^f < S_{i+1} \leq 1 \\
\lambda^{i+\frac{1}{2}}_w = \lambda_w(S_{i+1}) \text{ and } \lambda^{i+\frac{1}{2}}_n = \lambda_n(S_{i+1}), & 0 \leq S_{i+1} \leq S_{i+1}^f
\end{cases}
\]
CHAPTER 4. TRUST-REGION NEWTON FOR FLOW AND TRANSPORT

Note that we have $S^f=1$ for cocurrent flow, and $0 < S^f < 1$ for counter-current flow. Eqn. 4.9 implies that when $N_g < 0$, the wetting-phase flow direction does not change, and the flip of the nonwetting-phase flow direction depends only on whether $S_{i+1}$ (the saturation of the upwind cell w.r.t. $u_t$ for the downdip case) crosses $S^f=1$.

On the other hand, when $N_g > 0$ (updip), $u_t > 0$. Then, $\theta_n$ is always positive (Eqn. 4.7). From Eqn. 4.6, we get

$$
\begin{align*}
\lambda_{i+\frac{1}{2}} &= \lambda_w(S_i) \quad \text{and} \quad \lambda_{n+\frac{1}{2}} = \lambda_n(S_i), \quad 0 \leq \theta_w \leq \theta_n, \\
\lambda_{i+\frac{1}{2}} &= \lambda_w(S_{i+1}) \quad \text{and} \quad \lambda_{n+\frac{1}{2}} = \lambda_n(S_i), \quad \theta_w \leq 0 \leq \theta_n.
\end{align*}
$$

Then, there exists a unique $0 \leq S^f < 1$, such that $u_t = k g \Delta \rho \lambda_n(S_i)$, and thus $\theta_w = 0$. $S^f=0$ is 0 for cocurrent flow, and is $0 < S^f < 1$ for counter-current flow. Similar to the derivation in [44], we can write Eqn. 4.10 in explicit form as

$$
\begin{align*}
\lambda_{i+\frac{1}{2}} &= \lambda_w(S_i) \quad \text{and} \quad \lambda_{n+\frac{1}{2}} = \lambda_n(S_i), \quad S^f=0 < S_i \leq 1, \\
\lambda_{i+\frac{1}{2}} &= \lambda_w(S_{i+1}) \quad \text{and} \quad \lambda_{n+\frac{1}{2}} = \lambda_n(S_i), \quad 0 \leq S_i \leq S^f=0
\end{align*}
$$

Thus, when $N_g > 0$, the nonwetting-phase direction will not flip and the wetting-phase direction flips when the saturation value at cell $i$ (the upwind cell w.r.t. $u_t$ for this updip case) crosses $S^f=0$.

Combining the analyses for both cases ($N_g < 0$ and $N_g > 0$), we have completed the proof for the proposition that the phase-flow direction flips only when the saturation at the upwind cell w.r.t. total-velocity crosses $S^f=1$, or $S^f=0$.

Proposition 4.1.1 pinpoints the ‘flipping points’ for which flow reversal takes place. This information is used to overcome the convergence difficulties associated with the sequential-implicit solution strategy. Specifically, when the saturation solution update crosses $S^f=1$, or $S^f=0$, flow reversal is detected, and we change from the SIM to the fully-implicit method (FIM). This is the main idea of the nonlinear solver presented.
in the following section.

4.2 SIM→FIM

Now, we present a nonlinear solver to resolve the convergence difficulties associated with the sequential-implicit method (SIM) due to flow reversal. The simulation starts with the SIM. Within each outer loop, we solve the nonlinear flow problem (referred to as the pressure loop), then we update the total velocity, $u_t$, using the computed pressure field. We then calculate the ‘flipping points’ ($S_f=1$, or $S_f=0$) at each cell interface before solving the transport problem. Note that the ‘flipping points’ are the saturations where flow reversal occurs, and their values are determined by the sign of $u_t$ at each cell interface. Then, in the transport solver (saturation loop), the occurrence of flow reversal can be detected by checking whether the current update of the saturation at the upwind cell w.r.t. $u_t$ crosses a ‘flipping point’. If it does, flow reversal occurs. Once flow reversal is detected, the nonlinear solver performs the following steps: (1) chop the saturation update back to the ‘flipping point’, (2) stop the SIM loop, and (3) switch to FIM using the latest pressure and saturation fields as initial guesses. The FIM iterations are then performed until convergence for the current timestep is achieved. For the next timestep, the nonlinear solver is restarted using SIM. The algorithm for this nonlinear solution strategy - SIM→FIM - is summarized in Fig. 4.3.

In Fig. 4.3, we utilize the trust-region Newton scheme in the transport loop of SIM. The transport solution is thus unconditionally convergent given a fixed (in time) total-velocity field. However, the unconditional convergence of the transport solver cannot guarantee convergence of the outer loop of SIM. We expect that if SIM→FIM, then SIM can deal with the convergence difficulties other than flow reversal, which is resolved by FIM. Note that in FIM, we also employ the trust-region Newton scheme.
CHAPTER 4. TRUST-REGION NEWTON FOR FLOW AND TRANSPORT

Figure 4.3: The flow chart of SIM→FIM, which accommodate the coupling of viscous, gravitational, for one timestep.
CHAPTER 4. TRUST-REGION NEWTON FOR FLOW AND TRANSPORT

Since the total-velocity direction may change between FIM iterations, the trust-region scheme cannot guarantee convergence. However, numerical examples (e.g., examples in Section 3.7 and 4.3) indicate that the trust-region scheme improves FIM convergence for coupled flow and transport problems quite substantially.

As illustrated in the following numerical examples, this new nonlinear solver yields better convergence behavior than FIM alone for buoyancy-dominated problems with aggressive timestep sizes. Thus, the new nonlinear is a ‘smarter’ FIM with a better initial guess compared with the standard FIM.

4.3 Numerical examples

4.3.1 Simple 3×3 example

First, we use a simple case to illustrate how SIM→FIM detects flow reversal, triggers FIM, and enhances the convergence behavior of the nonlinear solver. Fig. 4.4 shows a 3×3 case under gravity segregation in a heterogeneous porous medium (log permeability is shown in Fig. 4.4(a)). The initial condition is \( S = 0.5 \) in the entire domain. Then, the wetting and nonwetting phases segregate with buoyancy as the only driving force. We simulate the segregation process with an aggressive timestep size (CFL≈308.8) using SIM→FIM. It is observed that no flow reversal occurs during the first 3 sequential outer-loop iterations. The wetting phase flows down and the nonwetting phase flows up at each cell interface. In the 4th outer-loop iteration, flow reversal occurs within the transport inner loop. To illustrate how flow reversal is detected for the 4th outer-loop iteration in Fig. 4.4(b), we plot the ‘flipping points’, which are calculated based on the \( u_t \) field before solving the transport problem. In the transport inner loop, we check whether any saturation update at the upwind cell
w.r.t. \( u_t \) crosses the corresponding ‘flipping point’. The ‘flipping points’ are plotted in the upwind w.r.t. each \( u_t \), instead of on the interface. Notice that the changes in the saturation in some cells have no impact on flow reversal, e.g., \( S_{1,3} \), \( S_{2,2} \) and \( S_{3,1} \), since they are not upwind w.r.t. to any \( u_t \). In the transport inner loop, saturations do not cross ‘flipping points’ during the first five inner-loop iterations. But during the sixth inner-loop iteration, one of the saturation, i.e., \( S_{2,1} \), crosses a ‘flipping point’ and flow reversal occurs. Fig. 4.4(c) shows how saturations change from the 5th to the 6th iteration. It is observed that \( S_{2,1} \) increases from \( S = 0.9950 \) to \( S = 0.9984 \) and crosses its corresponding flipping point (\( S = 0.9960 \)). Hence, the wetting-phase direction switches at the interface between (3, 1) and (2, 1), from flowing into cell (2, 1) to flowing out of cell (2, 1). The saturation change leads to flipping from counter-current to cocurrent flow. This case corresponds to type (d) in Fig. 4.2.

Intuitively speaking, the change in the wetting-phase direction (from downward to upward) at the interface between (3, 1) and (2, 1) can be caused by a decrease in \( S_{3,1} \), or an increase in \( S_{2,1} \). Both \( S_{3,1} \) and \( S_{2,1} \) increase in this example. Thus, we can interpret the change in the wetting-phase direction as caused by an increase in \( S_{2,1} \), not by a change in \( S_{3,1} \). This is consistent with the statement in proposition 4.1.1.

If we apply the standard SIM to simulate this case, there is no change in the vertical phase-flow direction during the first three outer-loops. Within the fourth outer-loop iteration, the phase-flow direction between cells (2, 1) and (3, 1) obtained from the pressure solution will change once the saturation solution is implicitly updated in the transport inner loop, and flow reversal occurs. If we stick with the SIM scheme after this flow reversal, the phase-flow directions change back once the pressure solution is updated by solving the flow problem. The phase-flow directions keep changing back and forth between the sequential flow and transport solutions from one outer-loop iteration to the next. As a result, the SIM scheme never converges for this case.
Figure 4.4: Numerical results with SIM→FIM: 2D incompressible two-phase gravity segregation problem for a $3 \times 3$ case (Initially wetting saturation is homogeneous in the domain $S = 0.5$), $M^d = 1$: (a) log permeability distribution; (b) flipping points at the upwind cell of $u_i$; (c) $S_{2,1}$ crossing the corresponding flip point ($S = 0.9960$) and hence wetting-phase direction flipping.
On the other hand, if we simulate this case using our SIM→FIM method, then there is no change in the vertical phase-flow direction during the first three outer-loops. Within the fourth outer-loop iteration, the ‘flipping points’ (as shown in Fig. 4.4(b)) are calculated before the transport loop. And in the transport loop, we detect flow reversal (saturation updates crossing the ‘flipping points’) at the interface between (3, 1) and (2, 1) (as shown in Fig. 4.4(c)), and we chop the saturation solution back to the ‘flipping points’. From this point on, we stop the SIM iterations and start FIM using latest pressure and saturation fields as initial guess until convergence is achieved. The SIM→FIM scheme takes four outer-loop sequential iterations and eight fully-implicit iterations to converge.

For this case, the FIM method with the standard Newton solver does not converge and takes 12 iterations to converge if our trust-region Newton scheme is used. For SIM→FIM, the four sequential iterations before switching to FIM resolve nonlinearities other than flow reversals, and they appear to provide an initial guess that allows the fully-implicit iterations to converge faster than the standard FIM approach.

In this simple case, we have illustrated how SIM→FIM detects flow reversal and switches to FIM. Next, we demonstrate the effectiveness of this nonlinear solver for several 2D incompressible two-phase examples with strong buoyancy.

### 4.3.2 2D examples

In this section, we test the performance of the SIM→FIM solver using several 2D examples. All the numerical examples use the same 2D heterogeneous model. It is a $20 \times 20$ domain, and the log permeability field is displayed in Fig. 4.5. We employ different reservoir settings and initial conditions. For each case, we compared the nonlinear performance of SIM, FIM with standard Newton, FIM with trust-region Newton, and our new nonlinear solver SIM→FIM.
In this case, the two phases (i.e., wetting and nonwetting) are segregated with buoyancy as the only driving force. The wetting phase is assumed to be heavier than the nonwetting phase, and initially the wetting phase is on top and the nonwetting phase is in the bottom (as shown in Fig. 4.6(a)). The boundaries are closed and the average wetting-phase saturation is \( S = 0.5 \). The end-point mobility ratio between the wetting and nonwetting phases is 1, i.e., \( M^0 = 1 \). We simulate this case for up to 5000 days. The numerical solution displayed in Fig. 4.6(b) corresponds to a small timestep size (\( \Delta t = 10, \text{CFL}=15 \)) and that shown in Fig. 4.6(c) is obtained with a large timestep size (\( \Delta t = 5000, \text{CFL}=765 \)). The ratio between two timestep sizes is 500. At the end of the simulation, the two phases are segregated and approach an equilibrium stage with the wetting phase (heavier) occupying the bottom-half of the domain and the nonwetting phase (lighter) lying in the top-half. Comparing Fig. 4.6(b) and 4.6(c), it is observed that there is more numerical dispersion when the large timestep size is employed, as expected. In Fig. 4.6(b) there is a sharp interface between the two segregated phases, while in Fig. 4.6(c), where the final solution is
obtained in one timestep, the interface between the two phases is smeared.

![Numerical results with SIM→FIM: 2D incompressible two-phase gravity segregation problem](image)

Figure 4.6: Numerical results with SIM→FIM: 2D incompressible two-phase gravity segregation problem (Initially wetting phase on top and nonwetting phase at bottom), $M^0 = 1$: (a) initial condition; (b) final solution with $\Delta t = 10$ (CFL≈15); (c) final solution with $\Delta t = 5000$ (CFL≈765).

The performance for different solution methods is compared in Table 4.1. It is clear that SIM does not converge for any timestep size, $\Delta t$, larger than or equal to 10. In fact, for this case, SIM only converges for $\Delta t \leq 1$. FIM, without trust-region chopping, can only converge for $\Delta t \leq 10$. And with the help of trust-region chopping,
Table 4.1: Performance with SIM, FIM and SIM→FIM: 2D incompressible two-phase gravity segregation problem (Initially wetting phase on top and nonwetting phase at bottom)

<table>
<thead>
<tr>
<th>total time</th>
<th>Δt</th>
<th>CFL (max)</th>
<th>SIM no chop iter / ts</th>
<th>FIM no chop iter / ts</th>
<th>SIM→FIM iter / ts</th>
<th>SIM→FIM FIM ts / total ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>10</td>
<td>14.83</td>
<td>-</td>
<td>3.33</td>
<td>1.75</td>
<td>1.55</td>
</tr>
<tr>
<td>5000</td>
<td>100</td>
<td>78.68</td>
<td>-</td>
<td>4.76</td>
<td>1.02</td>
<td>4.64</td>
</tr>
<tr>
<td>5000</td>
<td>1000</td>
<td>332.18</td>
<td>-</td>
<td>13.4</td>
<td>1.2</td>
<td>16.2</td>
</tr>
<tr>
<td>5000</td>
<td>5000</td>
<td>765.13</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>

FIM converges for timestep sizes up to 1000, but fails to converge for Δt = 5000. The chopping scheme improve the convergence performance of FIM quite substantially, but it cannot guarantee convergence. On the contrary, the new nonlinear solver, SIM→FIM can converge for all timesteps (up to Δt = 5000). For SIM→FIM, we count the number of SIM outer-loop iterations per timestep, the number of FIM iterations per timestep, and the ratio of the number of timesteps wherein FIM is triggered to the number of total timesteps. It is observed that when the timestep size is small (Δt = 10), flow reversal does not occur in every timestep. For less than half of the total timesteps (i.e., the ratio is 0.41), flow reversal occurs, and hence fully-implicit iterations are triggered. And when the timestep size is increased to 100 and beyond, FIM iterations get triggered for every timestep (the ratio is 1 in last column), revealing that flow reversal occurs in every timestep. For Δt = 5000 wherein FIM with chopping does not converge, the SIM→FIM runs two sequential outer iterations, and during the second iteration, flow reversal is detected. SIM is then stopped and FIM is started. Using the latest p and S from SIM as initial guesses, FIM converges in seven iterations. That is, for large timestep sizes, SIM→FIM has better convergence.
performance than FIM. This is because SIM, which fixes the total-velocity field for the transport inner loop, can take full advantage of the trust-region chopping scheme and drive the iterative solution to the true solution as much as possible before flow reversal occurs. Our experiments indicate that running SIM before FIM iterations can provide a better initial guess and hence improve the convergence performance of FIM.

**Lock exchange**

For this numerical example, the model setting is identical to the previous example except that the initial condition is such that the wetting (heavier) phase is in the right-half domain and nonwetting (lighter) phase is in the left-half domain and the end-point mobility ratio is $M^0 = 10$. The initial condition and the final solutions (with $\Delta t = 10$ and $\Delta t = 5000$) are displayed in Fig. 4.7. Due to the mobility difference between the two phases, the unfavourable displacement at the right-bottom quarter of the 2D domain (wetting displacing nonwetting) exhibits a much more smeared front than the one with favorable displacement in the left-top quarter (nonwetting displacing wetting). Also, by comparing Fig. 4.7(b) and 4.7(c), numerical diffusion due to the large timestep is present.

Table 4.2 shows the performance comparison between the different nonlinear solvers. For this case, SIM is again only convergent for $\Delta t \leq 1$ while FIM, even with the help of trust-region chopping, is only convergent for $\Delta t \leq 100$. That is, for this case, the convergence of FIM does not gain much from the trust-region chopping scheme. SIM→FIM can converge for all the timesteps we tested (up to $\Delta t = 5000$). For $\Delta t = 5000$, although SIM→FIM does take a large number of iterations (1 sequential and 128 fully-implicit iterations) to converge, it has better convergence behavior than SIM or FIM. That is, for this case, running just one SIM iteration before flow reversal can provide a good initial guess for FIM and accelerate the convergence of
Figure 4.7: Numerical results with SIM→FIM: 2D incompressible two-phase gravity segregation problem (Initially wetting phase at left and nonwetting phase at right): (a) initial condition; (b) final solution with $\Delta t = 10$ (CFL≈1.6); (c) final solution with $\Delta t = 5000$ (CFL≈389). $M^0 = 10$. 
FIM significantly.

Table 4.2: Performance with SIM, FIM and SIM→FIM: 2D incompressible two-phase gravity segregation problem (Initially wetting at left and nonwetting at right)

<table>
<thead>
<tr>
<th>total time</th>
<th>Δt</th>
<th>CFL (max)</th>
<th>SIM no chop</th>
<th>FIM no chop</th>
<th>SIM with chop</th>
<th>FIM with chop</th>
<th>SIM→FIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>10</td>
<td>1.61</td>
<td>-</td>
<td>3.32</td>
<td>3.61</td>
<td>2.15</td>
<td>2.17</td>
</tr>
<tr>
<td>5000</td>
<td>100</td>
<td>14.20</td>
<td>-</td>
<td>5.94</td>
<td>6.96</td>
<td>1</td>
<td>6.56</td>
</tr>
<tr>
<td>5000</td>
<td>1000</td>
<td>99.60</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>36.8</td>
</tr>
<tr>
<td>5000</td>
<td>5000</td>
<td>388.98</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>128</td>
</tr>
</tbody>
</table>

Quarter-five spot without gravity

In the previous two cases, we provide numerical validations of convergence performance for SIM→FIM in buoyancy-dominated problems. In this example, we show that in the absence of buoyancy forces, if there is no flow reversal occurring, the FIM iterations will never be triggered and SIM→FIM is the same as the SIM method. In this case, the permeability field is displayed in Fig. 4.5 and the initial condition is \( S = 0.0 \) (shown in Fig. 4.8(a)). A quarter-five spot well pattern is applied: the wetting phase is injected at the top-left corner and the nonwetting and wetting phases are produced at the bottom-right corner. Both the injector and producer are under the same total rate control. \( M^0 = 1 \). The final solutions are shown in Fig. 4.8(b) and 4.8(c).

The convergence performance of the different nonlinear solvers is reported in Table 4.3. SIM works very well for flow and transport problems under viscous forces. In fact, in the presence of viscous forces only, the phase flux in Eqn. 4.14 is independent of the total-velocity and hence independent of the pressure solution. In the special case, the
Figure 4.8: Numerical results with SIM→FIM: 2D incompressible two-phase quarter five-spot problem without gravity (Inject at top-left corner and produce at bottom-right corner): (a) initial condition; (b) final solution with $\Delta t = 10$ (CFL$\approx$6.3); (c) final solution with $\Delta t = 5000$ (CFL$\approx$1423). $M^0 = 1$. 
flow and transport problems are decoupled. Therefore, solving the flow and transport problems sequentially does not suffer from convergence difficulties. For this case, when the timestep size is large (\( \Delta t = 5000 \), CFL\( \approx 1423 \)), SIM takes 10 iterations to converge. On the other hand, without chopping, the nonlinear performance of FIM is so poor that it cannot converge for \( \Delta t \geq 10 \). With the help of trust-region chopping, the nonlinear convergence of FIM is improved quite significantly. In fact, it is convergent for all the tested timesteps (up to \( \Delta t = 5000 \)). In SIM→FIM, since flow reversal never occurs, the FIM iterations are not necessary, and hence FIM does not get triggered. This results in identical convergence performance (SIM iter/ts) between SIM→FIM and SIM. In this regard, SIM→FIM is reduced to SIM and still has a good convergence performance.

Table 4.3: Performance with SIM, FIM and SIM→FIM: 2D incompressible two-phase quarter five-spot problem without gravity (Inject at top-left corner and produce at bottom-right corner)

<table>
<thead>
<tr>
<th>total time</th>
<th>( \Delta t )</th>
<th>CFL (max)</th>
<th>SIM iter / ts</th>
<th>FIM iter / ts</th>
<th>FIM iter / ts</th>
<th>SIM→FIM iter / ts</th>
<th>SIM→FIM iter / ts</th>
<th>FIM ts / total ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>10</td>
<td>6.30</td>
<td>3.29</td>
<td>3.27</td>
<td>3.30</td>
<td>3.29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5000</td>
<td>100</td>
<td>56.65</td>
<td>-</td>
<td>4.76</td>
<td>4.24</td>
<td>4.76</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5000</td>
<td>1000</td>
<td>555.89</td>
<td>6.6</td>
<td>-</td>
<td>7.6</td>
<td>6.6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5000</td>
<td>5000</td>
<td>1422.5</td>
<td>10</td>
<td>-</td>
<td>12</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Quarter-five spot with gravity

We now consider a quarter-five spot problem under both viscous and buoyancy forces. The model setting for this case is identical to the previous one, except there is gravity
in the vertical direction. The final solutions with $\Delta t = 10$ and $\Delta t = 5000$, respectively, are shown in Fig. 4.9. The smeared displacement front in Fig. 4.9(b) is due to the truncation error associated with the large timestep. Comparing the solutions in Fig. 4.9 and Fig. 4.8 it is observed that in the presence of buoyancy forces, the non-wetting (lighter) phase flows upwards and is not displaced effectively by the injected wetting phase.

![Figure 4.9: Numerical results with SIM→FIM: 2D incompressible two-phase quarter five-spot problem with gravity (Inject at top-left corner and produce at bottom-right corner): (a) initial condition; (b) final solution with $\Delta t = 10$ (CFL≈8.3); (c) final solution with $\Delta t = 5000$ (CFL≈2924). $M^0 = 1$.](image)

We summarize the convergence performance of the different nonlinear solvers in Table 4.4. For this case, SIM cannot converge for $\Delta t \geq 10$. The standard FIM (without chopping) is convergent for $\Delta t \leq 10$. With the help of our trust-region chopping, FIM is able to converge for timesteps up to $\Delta t = 1000$. On the other hand, SIM→FIM is convergent for all the timesteps we tested, showing better convergence behavior than either SIM, or FIM, (with chopping).
Table 4.4: Performance with SIM, FIM and SIM→FIM: 2D incompressible two-phase quarter five-spot problem with gravity (Inject at top-left corner and produce at bottom-right corner)

<table>
<thead>
<tr>
<th>total time</th>
<th>( \Delta t )</th>
<th>CFL (max)</th>
<th>SIM</th>
<th>FIM</th>
<th>SIM→FIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>no chop</td>
<td>with chop</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SIM iter / ts</td>
<td>FIM iter / ts</td>
<td>FIM iter / ts</td>
</tr>
<tr>
<td>5000</td>
<td>10</td>
<td>8.32</td>
<td>-</td>
<td>3.32</td>
<td>3.61</td>
</tr>
<tr>
<td>5000</td>
<td>100</td>
<td>77.36</td>
<td>-</td>
<td>-</td>
<td>6.1</td>
</tr>
<tr>
<td>5000</td>
<td>1000</td>
<td>739.76</td>
<td>-</td>
<td>-</td>
<td>15</td>
</tr>
<tr>
<td>5000</td>
<td>5000</td>
<td>2923.89</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Chapter 5

Preconditioning for Nonlinear Solvers

In this chapter, we propose a preconditioning strategy to overcome convergence difficulties in the nonlinear solver that are associated with the propagation of saturation fronts into regions at or near the residual saturation. To illustrate this type of convergence difficulty, we report the performance of the nonlinear Newton-based solver for a simple 1D (1000 grid blocks) two-phase transport problem with only viscous forces. The initial condition in the domain is \( S = 0 \) (i.e., the nonwetting phase (oil) fully saturates the medium). The wetting phase \( (S = 1) \) is injected from the left boundary and we produce at constant total rate from the right boundary. The relative permeability functions are \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). The standard Newton is not convergent and the trust-region Newton scheme takes 262 iterations to convergence. In Chapter 3, we proved that the trust-region Newton algorithm is unconditionally convergent, but that proof did not deal with the convergence rate of the scheme. The 1D examples in Chapter 3 usually converge within 20 iterations for very large timestep sizes with the help of trust-region Newton scheme. The big difference in the number of iterations needed to achieve convergence (262 vs. 20) between the
1D examples in Chapter 3 and those shown in this chapter is due to the different initial and boundary conditions. In Chapter 3, the initial and boundary conditions are usually some distance away from the single-phase conditions, e.g., $S = 0.01$ for nonwetting phase and $S = 0.99$ for wetting phase. However, as described above, the initial and boundary conditions in this chapter are $S = 0$ and $S = 1$ respectively, which are strictly at single-phase conditions. As is discussed later in this chapter, only the (strictly) single-phase initial and boundary conditions (i.e., propagation of saturation fronts into regions at the residual saturation) will cause the above described convergence difficulties, whereas other initial and boundary conditions will not have such slow convergence.

To investigate the reason for this type of slow convergence, we plot the sequence of Newton updates and the final converged solution in Fig. 5.1. The initial guess is the initial condition, i.e., $S = 0$. Note that the Newton updates yield saturation distributions that are non-monotonic (green and red curves in Fig. 5.1), and then they ultimately converge to a discrete approximation that is monotone and accurate within a specified tolerance (light blue curve in Fig. 5.1). The non-monotonic ‘spikes’ in the saturation distribution start off being quite large, as shown in Fig. 5.2. In Fig. 5.2 it is noticed that the ‘spikes’ have very high saturation values, which are beyond the physical range $[0, 1]$ and hence correspond to non-physical mass accumulation. The ‘spikes’ are propagated downstream as the Newton iterations proceed. The propagation at the leading edge, where the injected fluid is invading a cell fully saturated with the resident fluid ($S = 0$), is constrained by the small (zero or near-zero) mobility of the invading phase in that cell. In fact, if the derivative of the relative permeability at $S = 0$ is zero, then the propagation proceeds one grid block per iteration. Ultimately, the saturation distribution becomes monotonic, and from that point on, the Newton updates converge quickly. So the non-monotonicity of the saturation solution is the reason for the slow convergence. The flux function for
Figure 5.1: Solutions for 1D transport problem with only viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). \( M = 1 \) and \( T = 100 \).

Figure 5.2: Spikes in solutions for 1D transport problem with only viscous forces. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). \( M = 1 \) and \( T = 100 \).
this case in Fig. 5.3. We can see that at $S = 0$, the slope of the flux function, i.e., $\frac{df}{dS}$, is zero. Recall that the slope of the flux function is the speed of the saturation wave. With $S = 0$ everywhere as the initial guess for the Newton solver, it is not possible to invade two successive cells in a single Newton update. This is because for each grid block, the influx comes from its upwind neighbor. If this immediate upwind cell has not been invaded by the injected wetting phase, the influx of the current grid block is zero and mass is transported one block per iteration. If the injected wetting phase for the current timestep cannot be transported beyond a single cell, then the mass associated with the timestep is placed in that cell. This explains the ‘spikes’ in the saturation solution during Newton iterations.

Since the convergence difficulties studied here are caused by the low wave speed at the leading edge, we propose a strategy that overcomes this constraint. For a given balance of forces (i.e., viscous, buoyancy, and capillary), the shape of the flux function is determined primarily by the relative permeability curves. Next, we analyze the wave speeds for linear relative permeability curves and then we consider more general nonlinear functions.
5.1 Linear relative permeability curves

Consider a one-dimensional horizontal model and assume that linear relative permeability curves are used, i.e.,

\[ k_{rw} = S, \quad (5.1) \]

and

\[ k_{ro} = 1 - S. \quad (5.2) \]

We denote the viscosity ratio as

\[ M = \frac{\mu_o}{\mu_w}. \quad (5.3) \]

If only viscous forces are present (e.g., water flooding in a horizontal domain), the wetting-phase flux can be written as

\[
\begin{align*}
  f &= \frac{k_{rw}/\mu_w}{k_{rw}/\mu_w + k_{ro}/\mu_o} \\
  &= \frac{S/\mu_w}{S/\mu_w + (1 - S)/\mu_o} \\
  &= \frac{MS}{1 + (M - 1)S}. \quad (5.4)
\end{align*}
\]

For this simple 1D setting, the residual equation for cell \( i \) in discretized form can be written as

\[ R = S_i^{n+1} - S_i^n + c \left( f_{i+1/2}^{n+1} - f_{i-1/2}^{n+1} \right), \quad (5.5) \]

where \( c \) is a dimensionless timestep

\[ c = \frac{u_t \Delta t}{\Delta x}. \quad (5.6) \]
Since only viscous forces are present, the flux at the right and left interfaces are

\[ f_{i+1/2}^{n+1} = \frac{M S_{i}^{n+1}}{1 + (M - 1) S_{i}^{n+1}}, \]  
(5.7)

and

\[ f_{i-1/2}^{n+1} = \frac{M S_{i-1}^{n+1}}{1 + (M - 1) S_{i-1}^{n+1}}. \]  
(5.8)

Hence, the residual can be expressed as

\[
R = S_{i}^{n+1} - S_{i}^{n} + c \left( f_{i+1/2}^{n+1} - f_{i-1/2}^{n+1} \right)
= S_{i}^{n+1} - S_{i}^{n} + cM \left( \frac{S_{i}^{n+1}}{1 + (M - 1) S_{i}^{n+1}} - \frac{S_{i-1}^{n+1}}{1 + (M - 1) S_{i-1}^{n+1}} \right)
= (S_{i}^{n+1} - S_{i}^{n}) + cM \left( \frac{S_{i}^{n+1}}{1 + (M - 1) S_{i}^{n+1}} - \frac{S_{i-1}^{n+1}}{1 + (M - 1) S_{i-1}^{n+1}} \right)
= (S_{i}^{n+1} - S_{i}^{n}) + cM \frac{S_{i}^{n+1} - S_{i-1}^{n+1}}{(1 + (M - 1) S_{i}^{n+1}) (1 + (M - 1) S_{i-1}^{n+1})}.
\]  
(5.9)

We obtain

\[
\frac{\partial R}{\partial S_{i}^{n+1}} = 1 + \frac{cM}{(1 + (M - 1) S_{i}^{n+1})^2},
\]  
(5.10)

and

\[
\frac{\partial R}{\partial S_{i-1}^{n+1}} = \frac{-cM}{(1 + (M - 1) S_{i}^{n+1}) (1 + (M - 1) S_{i-1}^{n+1})}.
\]  
(5.11)

If the initial condition is \( S^n = 0.0 \) and we take it as initial guess for solution at \( n + 1, \)
then
\[ \frac{\partial R}{\partial S_{n+1}^i} = 1 + cM, \] (5.12)
and
\[ \frac{\partial R}{\partial S_{n+1}^{i-1}} = -cM. \] (5.13)

For Newton method, the Jacobian for the first iteration is
\[
\begin{bmatrix}
1 + cM \\
-cM & 1 + cM \\
-cM & 1 + cM \\
& \ddots & \ddots \\
\end{bmatrix}.
\] (5.14)

Hence the timestep size \( c \) is reflected in the wave speed. See Russell [7]. Assuming that the left-boundary condition is \( S = S_0 \), the residual (right hand side) for the first iteration is:
\[
\begin{bmatrix}
-cM \\
\frac{S_0}{1+(M-1)S_0} \\
0 \\
\vdots \\
0
\end{bmatrix}.
\] (5.15)

Other boundary conditions result in different value for the residual term of the leftmost cell, whereas the residual terms of other cells will remain zero at the first iteration, regardless of the form of the boundary condition. Thus, starting from the initial guess, \( S = 0 \), the non-zero term in the residual is propagated downstream and the saturation is dispersive throughout the domain, since there is a non-zero element in every row of the lower off-diagonal part of the Jacobian (Eq. 5.14).
Specially, when \( M = 1 \), the Jacobian has the following form
\[
\begin{bmatrix}
1 + c \\
-c & 1 + c \\
-c & 1 + c \\
\vdots & \ddots & \ddots
\end{bmatrix}.
\] (5.16)

In fact, for \( M = 1 \), the residual equation, \( R \), becomes a linear function of the solution,
\[
R = S_i^{m+1} - S_i^m + c(S_i^{m+1} - S_{i-1}^{m+1}),
\] (5.17)
and this case, the Newton method will converge in one iteration regardless of the timestep size.

## 5.2 More general relative permeability curves

Assume the relative permeability curves are given by
\[
k_{rw} = S^\alpha,
\] (5.18)
and
\[
k_{ro} = (1 - S)^\beta.
\] (5.19)
where \( \alpha \geq 1 \) and \( \beta \geq 1 \). Then, the flux function can be written as
\[
f = \frac{S^\alpha/\mu_w}{S^\alpha/\mu_w + (1 - S)^\beta/\mu_o}.
\] (5.20)
Hence, the derivative is
\[
\frac{df}{dS} = \frac{\alpha S^\alpha \left( \frac{S^\alpha}{\mu_w} + (1-S)^\beta \right) - \alpha S^{\alpha-1} \left( \frac{S^{\alpha-1}}{\mu_w} - \beta (1-S)^{\beta-1} \right)}{\left( \frac{S^\alpha}{\mu_w} + (1-S)^\beta \right)^2}
\]
\[
= \frac{(\alpha S^{\alpha-1}(1-S)^\beta + \beta S^\alpha (1-S)^{\beta-1})}{\mu_w \mu_o \left( \frac{S^\alpha}{\mu_w} + (1-S)^\beta \right)^2}
\]
\[
= \frac{(\alpha S^{\alpha-1}(1-S)^\beta + \beta S^\alpha (1-S)^{\beta-1})}{M \left( \frac{S^\alpha}{M} + (1-S)^\beta \right)^2}. \quad (5.21)
\]

For \( \alpha > 1 \), \( \frac{df}{dS} = 0 \) for \( S = 0.0 \).

The residual is expressed as
\[
R = S^{n+1}_i - S^n_i + c \left( f^{n+1}_{i+1/2} - f^{n+1}_{i-1/2} \right). \quad (5.22)
\]

Hence
\[
\frac{\partial R}{\partial S^{n+1}_i} = 1 + c \frac{df^{n+1}_{i+1/2}}{dS^{n+1}_i}, \quad (5.23)
\]
and
\[
\frac{\partial R}{\partial S^{n+1}_{i-1}} = -c \frac{df^{n+1}_{i-1/2}}{dS^{n+1}_{i-1}}. \quad (5.24)
\]

Assume \( \alpha > 1 \). It follows that when \( S^n = 0 \) everywhere and is taken as the initial guess for the next timestep \( n + 1 \), then \( \frac{\partial R}{\partial S^{n+1}} = 1 \) and \( \frac{\partial R}{\partial S^{n+1}_{i-1}} = 0 \). So, the Jacobian matrix for the first Newton iteration is the identity matrix.
\[
\begin{bmatrix}
1 \\
0 & 1 \\
0 & 1 \\
\cdot & \cdot & \cdot
\end{bmatrix}. \quad (5.25)
\]
The timestep size $c$ does not appear in the first Jacobian matrix. Assuming that $f = 1$ at the left boundary, the corresponding residual for the first iteration is

$$
\begin{bmatrix}
-c \\
0 \\
\vdots \\
0
\end{bmatrix}.
$$

Since the Jacobian matrix is the identity, we can see that the non-zero term in the residual cannot propagate more than one block after the first iteration.

On the other hand, even though $S^n = 0.0$ is the initial condition, if we take the initial guess of the solution for the current timestep, $S^{n+1,0}$, as $S^* \neq 0.0$, we then have

$$
\frac{\partial R}{\partial S_{i+1}^{n+1}} = 1 + cf^*,
$$

and

$$
\frac{\partial R}{\partial S_{i+1}^{n+1}} = -cf^*.
$$

Hence, the Jacobian matrix (i.e., $J$) for the first iteration is

$$
\begin{bmatrix}
1 + cf^* \\
-cf^* & 1 + cf^* \\
-cf^* & 1 + cf^* \\
\vdots & \ddots & \ddots
\end{bmatrix};
$$

where $f^*$ is $\frac{df}{ds}$ evaluated at $S^*$. The timestep size $c$ appears in the Jacobian matrix if $f^* \neq 0$. Assuming that the left boundary condition is $f = 1$, the corresponding
residual vector for the first iteration is

\[
\begin{bmatrix}
S^* + c(f^* - 1) \\
S^* \\
\vdots \\
S^*
\end{bmatrix}
\]  
(5.30)

the solution update, \( \delta S \), is obtained by solving the linear system \( J \delta S = -R \), and then \( S^* + \delta S \) serves as the starting point for the second Newton iteration.

Now we prove that for the first Newton iteration, if \( f''^* > 0 \), the resulting \( S^* + \delta S \) of our 1D problem is decreases monotonically as \( i \) increases from 1 to \( N \), where \( N \) is the number of blocks.

**Proof.** Since the Jacobian matrix [5.29] is lower triangular, we can solve the elements in \( \delta S = [\delta S_1, \delta S_2, \cdots, \delta S_N]^T \) one by one.

\[
\begin{bmatrix}
1 + cf'^* \\
-cf'^* & 1 + cf'^* \\
-cf'^* & 1 + cf'^* & \cdots & \cdots \\
\vdots & \vdots & \ddots & \ddots \\
\end{bmatrix}
\begin{bmatrix}
\delta S_1 \\
\delta S_2 \\
\vdots \\
\delta S_N
\end{bmatrix}
= -
\begin{bmatrix}
S^* + c(f^* - 1) \\
S^* \\
\vdots \\
S^*
\end{bmatrix}
\]  
(5.31)

First, we obtain the solution update for the first block. Namely,

\[
\delta S_1 = - \frac{S^* + c(f^* - 1)}{1 + cf'^*}.
\]  
(5.32)
Hence

\[ S^* + \delta S_1 = S^* - \frac{S^* + c(f^* - 1)}{1 + cf^{*}} \]
\[ = \frac{S^* + cS^* f^{*} - S^* - cf^*}{1 + cf^{*}} \]
\[ = \frac{c(S^* f^* - f^* + 1)}{1 + cf^{*}} \]
\[ > 0. \quad (5.33) \]

For \( i \geq 2 \)

\[ - cf^{*} \delta S_{i-1} + (1 + cf^{*}) \delta S_i = -S^*. \quad (5.34) \]

Therefore,

\[ \delta S_i + S^* = \frac{cf^{*}}{1 + cf^{*}} (\delta S_{i-1} + S^*), \quad (5.35) \]

and since \( f^{**} > 0 \),

\[ 0 < \frac{cf^{*}}{1 + cf^{*}} < 1. \quad (5.36) \]

It follows that

\[ \delta S_1 + S^* < \delta S_{i-1} + S^*. \quad (5.37) \]

Then, we have proved that \( S^* + \delta S \) decreases monotonically as \( i \) increases.

The solution can be written as

\[ \delta S_i + S^* = \left( \frac{cf^{*}}{1 + cf^{*}} \right)^{i-1} (\delta S_1 + S^*), \quad (5.38) \]

and since \( S^* + \delta S_1 > 0 \) (Eqn. 5.33), then

\[ \delta S_i + S^* > 0 \quad \forall i \geq 2. \quad (5.39) \]
That is, the saturation solution from the first iteration is positive (and less than unity) in every block of the 1D domain.

From Eqn. 5.35 we can see that for any initial guess, $S^*$, if $f'^* > 0$, the first Newton iteration yields a saturation distribution that is monotonic and positive. And to allow for maximum propagation of the saturation waves downstream, we make $\delta S_i + S^*$ as close to $\delta S_{i-1} + S^*$ as possible. Thus, we maximize $cf'^*$. Specifically, we use the maximum $f'^*$, i.e., we use the inflection point. Next, we propose a preconditioning strategy for nonlinear transport solvers based on this analysis.

### 5.3 Preconditioning strategy

In the previous section, we showed that when the relative permeability curves are not linear (i.e., $\alpha > 0$) and the initial condition is single-phase, the convergence difficulties associated with the existing nonlinear solutions are due to the low wave propagation speed at the leading edge where the injected fluid is about to invade a single-phase cell. Recall that the speed is represented by the slope $\left(\frac{df}{dS}\right)$ of the flux function. The mass in the grid block being invaded by the injected fluid is moving much slower than the mass in cells further upstream and that leads to unphysical mass accumulation. These ‘spikes’ in the saturation distribution obtained from the Newton sequence can slow convergence substantially. In fact, if the timestep is quite large, the Newton method fails to converge at all. We propose a preconditioning strategy to resolve this particular convergence problem. The idea is to provide an initial guess for the Newton solver that has high speeds of wave propagation such that the saturation waves can spread throughout the domain in the first Newton iteration. That is, we spread the mass associated with the timestep as far downstream as needed to ensure that the saturation field has no ‘spikes’. To obtain the high speed of wave propagation, we use
the inflection point of the flux function as the initial guess for every grid block. The inflection point the largest slope (Fig. 5.3); hence, mass in every grid block moves no slower than the mass in the upwind grid blocks.

Note that when the buoyancy is significant, the flux function is not monotonic, and there are two inflection points on the flux function (Fig. 1.1(b)). In such cases, we use $S_{\text{inflec1}}$, the inflection point smaller than $S_f^1$, as the initial guess for the grid blocks in the imbibition region (wetting phase displacing nonwetting phase), and we use $S_{\text{inflec2}}$, the inflection point larger than $S_f^1$, as the initial guess for the grid blocks in the drainage region (nonwetting phase displacing wetting phase).

Thus, the preconditioning strategy provides a ‘smarter’ initial guess for the nonlinear solver that overcomes limitations due to slow wave propagation in the domain. If the flux function is known (for example, in trust-region Newton method, we already calculated the inflections points), there is no additional overhead cost associated with applying our preconditioning strategy.

The preconditioning strategy, which provides an initial guess for the first Newton iteration, can be applied for 1D transport in a straightforward way. For example, for the 1D transport problem discussed at the beginning of this Chapter, without preconditioning, the nonlinear solver converges within 262 iterations while with preconditioning, the nonlinear solver is able to converge within 7 iterations. That is, we can save more than one order of magnitude in the computational cost by using our preconditioning strategy. To illustrate how preconditioning accelerates the convergence rate, we plot the initial guess, the sequence of iterative updates and the converged solution for the trust-region Newton method in Fig. 5.4. The saturation distribution during the iterative process is shown, respectively, for both without and with preconditioning strategy.

Without preconditioning, the initial guess for the nonlinear solver is the same as the initial condition, i.e., $S = 0$. We observe in Fig. 5.4(a) that, if no preconditioning
is employed, there are ‘spikes’ in the iterative solutions. The ‘spikes’ move slowly as the Newton iterations increase. Hence, the nonlinear solver consumes a significant amount of iterations to converge. On the other hand, using the inflection point as the initial guess for the first Newton iteration (blue dashed lines shown in Fig. 5.4(b)) leads to fast convergence. With preconditioning, the first iterative solution (green solid line) is monotonic, and there are no ‘spikes’ in the solution. Mass in every grid block is moving no slower than the mass flow of upwind grid block, and there are no ‘spikes’ due to unphysical mass accumulation. During the following iterations, the saturation distribution gets sharpened and converges quickly to the solution.

To handle coupled flow and transport in 2D or 3D domains, the preconditioning strategy can be used with either sequential-implicit or fully-implicit solution schemes.

**Sequential-implicit method:** The sequential-implicit method treats the flow (pressure) and transport (saturation) separately and solves them in a sequential manner. There is an outer loop to control the splitting error. In the outer loop, there is a flow (pressure) nonlinear loop and a transport (saturation) nonlinear loop. We use the preconditioning strategy with the sequential-implicit method by providing an
initial guess for the nonlinear transport (saturation) loop. This assumes that the flow loop yields a well behaved pressure field and that the total-velocity is calculated and fixed when the transport loop is entered. Note that we provide the initial guess for the transport loop for the first iteration of the outer loop. For the subsequent outer iterations, given that the transport loop is convergent, the fluid in every grid block such that the fluid at an upstream cell moves no faster than that of a downstream cell.

**Fully-implicit method:** The fully-implicit method (FIM) solves for the pressure and saturation fields simultaneously. Since the preconditioning strategy only provides initial guesses for the saturation solution, when it is used in the fully-implicit method, only the initial guesses for saturation variables are modified to be the inflection point, the initial guesses for the pressure variables are unaffected. Note that since the initial pressure field can be quite different from the converged solution, and the pressure and saturation are updated simultaneously, we will need to apply the preconditioning strategy twice for each time step. That is, we first apply the strategy at the beginning of each time step to provide initial guesses for saturation to obtain a reasonable pressure field, and then we apply the strategy again after the first Newton iteration where the pressure variables have already been updated. Consequently, the saturation after the first Newton iteration will be kept the same values as the initial guesses provided by the preconditioning strategy and updates to the saturation variables from the first Newton iteration will be effectively zero.

### 5.4 Numerical examples

In this section, we test the performance of the preconditioning strategy by applying it to the fully-implicit method. 2D and 3D reservoirs with highly heterogeneous permeability fields are tested. In addition, we also test the preconditioning strategy
for 1D transport problems and 2D flow and transport problems (simulated using sequential-implicit method). Corresponding performance results are summarized in Appendix E.

5.4.1 2D flow and transport

To test the performance of the preconditioning strategy for fully-implicit method, we embedded the strategy into AD-GPRS. [17, 28, 51]. First, we test a wetting-phase flooding problem in a 2D domain, i.e., the bottom layer of SPE 10. The permeability field is displayed in Fig. 5.5. The initial condition is $S = 0.15$, which is the end-point saturation. Wetting phase is injected at right-top corner, $(220, 1)$, and there is a producer at left-bottom corner, $(1, 60)$. The injector is under rate control whereas the producer is under BHP control. The densities of the nonwetting phase and wetting phase, respectively, are $120.5 \text{ kg/m}^3$ and $1009.3 \text{ kg/m}^3$. Thus buoyancy forces cannot be ignored. The two-phase relative permeability functions are from the SPE 9 model [53], as shown in Fig. 3.37. The simulation ends at $T = 100$ days, which corresponds to $0.83$ PVI. The timestep size is fixed at $\Delta t = 100$ days, so that the simulation is performed in a single timestep. This is a highly heterogeneous (channelized) reservoir,

![Figure 5.5: Permeability of SPE 10 bottom layer](image-url)
and the CFL number varies by several orders of magnitude in the reservoir model. The maximum CFL number is 392.48, which is quite challenging for nonlinear solvers. Fig. 5.6 shows the sorted CFL numbers for this wetting-phase flooding problem. In fact, more than 56% of the cells have CFL numbers larger than unity, and around 6% cells have CFL numbers larger than 10.

Figure 5.6: CFL distribution for two phase flow and transport in top 10 layers of SPE 10 model ($60 \times 220 = 13,200$), in the presence of buoyancy forces. End of simulation: $T = 100$ days (0.83% PVI). Timestep size: $\Delta t = 100$. Maximum CFL=392.48.

We simulate the same case with different nonlinear solvers: standard Newton, standard Newton with preconditioning, Newton with Appleyard chopping, Newton with Appleyard chopping and preconditioning, trust-region Newton, trust-region Newton with preconditioning. The performance of these six nonlinear solvers is summarized in Table 5.1. In order to provide a more illustrative comparison, we plot the total number of Newton iterations for the six nonlinear solvers in Fig. 5.7 and the total simulation time in Fig. 5.8 in logarithmic scale.

We use a fixed timestep size $\Delta t = 100$. If convergence is not achieved within 30 iterations for the target timestep size, we cut the timestep size by half. From Table 5.1 and Fig. 5.7 we can see that both the standard Newton method and Newton with Appleyard chopping take significant amounts of total iterations (i.e., 9525 and 1715)
to converge, due to multiple timestep cuts and wasted iterations. On the contrary, the trust-region method improves the nonlinear performance and it can converge in 74 total iterations with only one timestep cut. With preconditioning, the standard Newton method converges within 3075 iteration and Newton with Appleyard chopping is able to converge within 661 iterations. The preconditioning strategy improves the nonlinear convergence for both nonlinear solvers. Specially, with the help of the preconditioning, the trust-region Newton method can converge in 25 iterations, and there are no timestep cuts.

Table 5.1: Summary of performance for two phase flow and transport in bottom layer of SPE 10 model ($60 \times 220 = 13,200$), in the presence of buoyancy forces. End of simulation: $T = 100$ days (0.83% PVI). Timestep size: $\Delta t = 100$ days. Maximum CFL=392.48. ‘Wasted timesteps and ‘wasted iterations indicate the number of timesteps and Newton iterations that are wasted due to timestep cuts.

<table>
<thead>
<tr>
<th></th>
<th>standard</th>
<th>Appleyard</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>no precond.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timesteps</td>
<td>71</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>iterations</td>
<td>368</td>
<td>364</td>
<td>43</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>369</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>9157</td>
<td>1351</td>
<td>31</td>
</tr>
<tr>
<td>total iterations</td>
<td>9525</td>
<td>1715</td>
<td>74</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>1362.01</td>
<td>274.07</td>
<td>66.42</td>
</tr>
<tr>
<td><strong>with precond.</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timesteps</td>
<td>34</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>iterations</td>
<td>290</td>
<td>201</td>
<td>25</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>116</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>2785</td>
<td>460</td>
<td>0</td>
</tr>
<tr>
<td>total iterations</td>
<td>3075</td>
<td>661</td>
<td>25</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>508.53</td>
<td>115.38</td>
<td>25.03</td>
</tr>
</tbody>
</table>

The difference in nonlinear performance between these nonlinear solvers results in the difference in the total simulation time (Table 5.1 and Fig. 5.8). The trust-region Newton method is more than 20 times faster than the standard Newton method. Moreover, the trust-region with preconditioning is around 55 times faster than the
standard Newton (without preconditioning). More significant difference in total simulation time can be observed with larger-scale models, as shown in the next example.

![Graph showing total number of (FIM) Newton iterations for two phase flow and transport in bottom layer of SPE 10 model (60 x 220 = 13,200). End of simulation: T = 100 days (0.83% PVI). Timestep size: \( \Delta t = 100 \). Maximum CFL=392.482.](image1)

![Graph showing total simulation time for two phase flow and transport in bottom layer of SPE 10 model (60 x 220 = 13,200). End of simulation: T = 100 days (0.83% PVI). Timestep size: \( \Delta t = 100 \). Maximum CFL=392.482.](image2)

To better understand the mechanism of our preconditioning strategy when used
in the fully-implicit method, we found that the ‘spikes’ in the iterative saturation solution, due to the slow wave speed and mass accumulation at the wave front, trigger dramatic updates in the pressure solution and lead to non-monotonic pressure fields prior to convergence. The reason is that for the fully-implicit method, where pressure and saturation are solved simultaneously, once the non-physical ‘spikes’ in the saturation field form and propagate, the pressure drops down dramatically in the cells around the ‘spikes’. In the next iteration, since the pressure within the ‘spikes’ is higher than that of the cells around it, the mass within the ‘spikes’ can be distributed to the neighbouring cells. The dramatic pressure update and non-monotonic pressure distribution can greatly slow down the convergence of the fully-implicit method. By applying preconditioning, we can avoid the non-physical ‘spikes’ in the saturation field and the non-monotonic pressure distribution. Therefore, the preconditioning strategy is able to accelerate the Newton convergence for the fully-implicit method quite significantly.

5.4.2 3D flow and transport

The test cases are two-phase wetting-phase flooding problems in the top 10 layers of the SPE 10 model with fully-implicit method. It is a three-dimensional, highly heterogeneous reservoir, containing $60 \times 220 \times 10 = 132,000$ grid blocks. Initially, the reservoir is fully saturated with the nonwetting phase ($S = 0.0$). We inject the wetting phase at one corner of the reservoir, i.e., $(1, 1, 1)$, and produce the wetting and nonwetting phases at another corner of the reservoir, i.e., $(60, 220, 10)$. Both wells have a single completion. The injector is under a constant rate control ($10 m^3$/day).

Here, we test the wetting-phase flooding problem in the top 10 layers of SPE 10 model in the presence of buoyancy forces. The densities of the nonwetting phase and wetting phase, respectively, are $720.5 \ kg/m^3$ and $1009.3 \ kg/m^3$. We use the two-phase relative permeability curves as shown in Fig. 3.37. We simulate the transport
problem until \( T = 100 \) days, which corresponds to 0.364% PVI. Fixed timestep size is used: \( \Delta t = 100 \) days. We plot the sorted CFL numbers in Fig. 5.9. The CFL number varies by orders of magnitude. Around 1% of cells (i.e., 1320 cells) in the reservoir have CFL numbers larger than 10 and the maximum CFL number is 603.

![CFL distribution](image)

Figure 5.9: CFL distribution for two phase flow and transport in top 10 layers of SPE 10 model (60 \( \times \) 220 \( \times \) 10 = 132,000), in the presence of buoyancy forces. End of simulation: \( T = 100 \) days (0.364% PVI). Timestep size: \( \Delta t = 100 \). Maximum CFL=603.19.

We simulate this case using six different nonlinear solvers: standard Newton, standard Newton with preconditioning, Newton with Appleyard chopping, Newton with Appleyard chopping and preconditioning, trust-region Newton method, and trust-region Newton method with preconditioning. The comparison of performance between these nonlinear solvers is summarized in Table 5.2. We also plot the total number of iteration and total simulation time, respectively, in Fig. 5.10 and Fig. 5.11 for illustration (in logarithmic scale). From the comparison, we can see that the standard Newton and the Newton with Appleyard chopping take significant amounts of Newton iterations (respectively, 18750 and 4694) to converge, due to large numbers of timestep cuts and wasted iterations, and the trust-region Newton has superior convergence performance. It only takes 147 iterations and 3 timestep cuts to converge.
The reduction in the total Newton iterations by more than an order of magnitude results in a corresponding reduction in the overall computational cost. The trust-region Newton method is around 10 times faster than the Newton method with Appleyard chopping, and more than 37 times faster than the standard Newton.

Moreover, we can achieve higher reduction in the total Newton iterations, as well as, the computational cost by using the preconditioning strategy, as shown in Table 5.2. With preconditioning, the standard Newton takes 4219 iterations, Newton with Appleyard takes 1229 iterations and the trust-region Newton with preconditioning takes 23 iterations to converge and there are no timestep cuts. That is, the preconditioning strategy improves the nonlinear performance of all three nonlinear solvers significantly (standard Newton: 18750 vs. 4219, Newton with Appleyard chopping: 4694 vs. 1229, and trust-region Newton: 147 vs. 23). We can see that in the presence of buoyancy forces, by applying the preconditioning strategy, the standard Newton still yields much inferior nonlinear performance to the trust-region Newton method. Since buoyancy forces cannot be ignored in this example, the saturation solution will have four trust regions, partitioned by two inflection points and one unit-flux point. By applying the preconditioning strategy and using one of the inflection points as the initial guess, it is unlikely that the iterative solutions will always reside in the same trust region as the true solution, since the iterative solutions can arbitrarily cross the trust-region boundaries. It is clear that the trust-region Newton method with preconditioning is the optimal choice, which reduces the number of Newton iterations significantly and thus yields great potential in the saving of the computational time. In terms of the total computational cost, trust-region Newton method with preconditioning is almost 60 times faster than Newton with Appleyard chopping (without preconditioning) and more than 200 times faster than the standard Newton method (without preconditioning).
CHAPTER 5. PRECONDITIONING FOR NONLINEAR SOLVERS

Table 5.2: Summary of performance for two phase flow and transport in top 10 layers of SPE 10 model ($60 \times 220 \times 10 = 132,000$), in the presence of buoyancy forces. End of simulation: $T = 100$ days (0.364% PVI). Timestep size: $\Delta t = 100$ days. Maximum CFL=603.19. ‘Wasted timesteps and ‘wasted iterations indicate the number of timesteps and Newton iterations that are wasted due to timestep cuts.

<table>
<thead>
<tr>
<th></th>
<th>standard</th>
<th>Appleyard</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td>no precond.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timesteps</td>
<td>113</td>
<td>33</td>
<td>2</td>
</tr>
<tr>
<td>iterations</td>
<td>574</td>
<td>813</td>
<td>54</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>706</td>
<td>151</td>
<td>3</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>18176</td>
<td>3881</td>
<td>93</td>
</tr>
<tr>
<td>total iterations</td>
<td>18750</td>
<td>4694</td>
<td>147</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>44558.65</td>
<td>12024.39</td>
<td>1204.02</td>
</tr>
<tr>
<td>with precond.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>timesteps</td>
<td>36</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>iterations</td>
<td>342</td>
<td>327</td>
<td>23</td>
</tr>
<tr>
<td>wasted timesteps</td>
<td>148</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>wasted iterations</td>
<td>3877</td>
<td>902</td>
<td>0</td>
</tr>
<tr>
<td>total iterations</td>
<td>4219</td>
<td>1229</td>
<td>23</td>
</tr>
<tr>
<td>simulation time (s)</td>
<td>11285.62</td>
<td>3622.33</td>
<td>210.25</td>
</tr>
</tbody>
</table>

Figure 5.10: Total number of (FIM) Newton iterations for two phase flow and transport in top 10 layers of SPE 10 model ($60 \times 220 \times 10 = 132,000$), in the presence of buoyancy forces. End of simulation: $T = 100$ days (0.364% PVI). Timestep size: $\Delta t = 100$. Maximum CFL=603.19.
Figure 5.11: Total simulation time for two phase flow and transport in top 10 layers of SPE 10 model ($60 \times 220 \times 10 = 132,000$), in the presence of buoyancy forces. End of simulation: $T = 100$ days (0.364% PVI). Timestep size: $\Delta t = 100$. Maximum CFL=603.19.
Chapter 6

Conclusions and Future Work

6.1 Summary and conclusions

In this dissertation, we developed new nonlinear solvers for two-phase flow and transport where viscous, buoyancy, and capillary forces are all significant.

First, we proposed a trust-region Newton solver to resolve the convergence difficulty for immiscible multiphase transport in porous media. The solver employs trust regions of the flux function to guide the Newton iterations. The delineation of the trust-region boundaries is detected by the unit-flux and inflection points of the flux function. The unit-flux point is the saturation value where the flux equals to 1, i.e., $f = 1$. The unit-flux and inflection points are available before solving the transport problem, since for any given total velocity, the flux function depends on the mobility ratio and the gravity number.

The trust-region nonlinear solver is essentially a multi-point chopping strategy. That is, at the end of each Newton iteration, we do not allow two successive updates to cross any of these critical points. We proved global convergence of our trust-region Newton method for single-cell transport problems. That is, convergence is guaranteed for arbitrary initial guesses and arbitrary timestep sizes. Then, numerical examples
including single-cell, two-cell, and multi-cell 1D problems that span the full range of
the parameter space of viscous, gravity and capillary forces, are used to demonstrate
the unconditional convergence of our trust-region Newton scheme.

Comparison between existing methods (standard Newton and Newton with Appleyard chopping) and the trust-region nonlinear solver for the simulation of two-phase flow and transport using the upper zone of the SPE 10 model (Tarbert formation) and the full SPE 10 model demonstrates that our new nonlinear solver exhibits superior convergence performance for large-scale, heterogeneous reservoir models. As opposed to worrying about the ability of the nonlinear solver to converge, one can choose the timestep based on accuracy requirements. Of course, this requires detailed analysis of the complex time-truncation errors associated with the nonlinear dynamics of multi-component, multiphase flow and transport in natural porous formations.

We also presented a nonlinear solver for coupled flow and transport problems in multi-dimensional domain. This effort is motivated by the convergence limitation associated with the sequential-implicit method when buoyancy is significant and counter-current flow occurs. When buoyancy forces are significant, the flow directions of the two phases - at the interface between neighbouring cells - based on the pressure solution can change as a function of iteration and time. Also, when the timestep size is relatively large, the phase-flow directions obtained from the pressure solution may change once the transport problem is solved for the saturation solution. And the updated phase-flow directions may change again once the pressure solution is solved within the next outer-loop iteration of the sequential approach. Changes in the phase-flow directions between the separate pressure and saturation solutions (or updates) can slow convergence of the outer-loop iteration quite significantly. In cases where aggressively large timesteps \( \text{CFL} \gg 1 \) are taken, the sequential-implicit scheme may not converge at all, thus requiring timestep cuts.

It was found that if there is no flow reversal during the transport inner loop,
convergence of the sequential-implicit method is guaranteed. We classified the flow reversal into four cases: cocurrent switching to counter-current due to nonwetting phase flipping, cocurrent switching to counter-current due to wetting phase flipping, counter-current switching to cocurrent due to nonwetting phase flipping, and counter-current switching to cocurrent due to wetting phase flipping. It was proved that for any of the four cases, flow reversal is caused by the saturation in the upwind cell of the total-velocity crossing the unit-flux point \( S_f^1 \), or the zero-flux point \( S_f^0 \).

The nonlinear solver proposed here detects flow reversal by checking whether the saturation update at the upwind cell of total-velocity crosses the unit-flux, or zero-flux, point. If flow reversal is detected, the sequential-implicit loop is terminated and the fully-implicit method is initiated using the latest non-converged solution of pressure and saturation as initial guesses. That is, we utilize the sequential-implicit method to resolve all the nonlinearities other than flow reversal and use fully-implicit method to handle the remaining part.

Numerical evidence indicates that the proposed nonlinear solver is able to converge for timesteps that are much larger than what the sequential-implicit method can handle. In addition, for very large timestep sizes, the new nonlinear solver yields better convergence performance than the fully-implicit method. This is due to the fact that the sequential-implicit method is able to take full advantage of the trust-region chopping scheme for the transport problem and hence provides better initial guesses than applying the fully-implicit method from the beginning of the timestep.

We also proposed a preconditioning strategy for nonlinear transport. For two-phase immiscible flow, slow convergence is observed if one-phase fluid is injected into the domain to displace the original fluid that is composed entirely of another phase. Numerical analysis reveals that the slow convergence is caused by the slow speed of wave propagation and the associated non-monotonicity of the saturation distribution during the Newton iterations. To resolve this issue, we proposed a preconditioning
strategy in which the initial guess for the Newton iteration is the inflection point, instead of the initial condition. Numerical examples of challenging transport problems show that the preconditioning strategy accelerates the convergence of existing nonlinear solvers quite significantly.

6.2 Recommendations for future work

The trust-region nonlinear solver can be further developed in the following directions.

- Two-phase multi-component flow. It is straightforward to extend the nonlinear solver to handle two-phase flow with mass transfer of the components across the fluid phases.

- Three-phase flow. For two-phase flow, the flux function depends on one saturation, while for three-phase flow the flux function depends on two saturations, and it can be represented by a curved surface. More complexities are involved and more efforts are required to investigate the parametrization of the flux surface before extending the trust-region method for three-phase flow.

- Time-truncation error. The trust-region method overcomes the convergence limitations associated with existing Newton-based nonlinear solvers. So, now one can choose the timestep size based on accuracy considerations only. In order to do this in a rigorous manner, we need to analyse the dependence of the truncation error on the timestep size. In theory, for first-order discretization scheme, the time truncation error depends on the timestep size. However, the coefficient of the dependence can vary quite significantly. That is, in practice, for a given timestep size, in different flow scenarios or different stages of the same scenario, the truncation errors can be very different.
• Coupling due to capillarity. The nonlinear solver for coupled flow and transport should also be extended to handle cases with strong capillary forces. The capillary pressure couples the flow and transport problems tightly. In the presence of strong capillarity, changes of the phase-flow direction between the separate pressure and saturation solutions (or updates) can slow convergence of the outer-loop iteration quite significantly. In cases where aggressively large timesteps ($\text{CFL} \gg 1$) are taken, the sequential-implicit scheme may not converge at all thus requiring timestep cuts. More efforts are needed to investigate the coupling between flow and transport problems in the presence of strong capillarity.
Nomenclature

\[ S \]  wetting phase saturation
\[ f \]  wetting phase flux, including viscous and buoyancy fluxes
\[ F \]  wetting phase flux, including viscous, buoyancy and capillary fluxes
\[ c \]  dimensionless timestep size
\[ S_{wr} \]  wetting (water) phase residual saturation
\[ S_{nwr} \]  non-wetting (oil) phase residual saturation
\[ p_c \]  capillary pressure
\[ P_e \]  Peclet number
\[ N_g \]  gravity number
\[ M^0 \]  end-point mobility ratio
\[ M \]  oil/water viscosity ratio
\[ C(S) \]  capillary function defined in Eqn. 3.4
\[ h \]  height (positive upwards)

Subscripts

\[ \alpha, \beta \]  phase
\[ n \]  nonwetting
\[ w \]  wetting
\[ t \]  total

Superscripts

\[ inflec \]  inflection point of the flux function
\[ sonic \]  sonic point of the flux function
\[ f = 1 \]  unit-flux point of the flux function
$f = 0$ zero-flux point of the flux function

$n$ previous time step

$\nu$ previous iteration

**Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIM</td>
<td>fully-implicit method</td>
</tr>
<tr>
<td>SIM</td>
<td>sequential-implicit method</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Friedrichs-Lewy number</td>
</tr>
<tr>
<td>CPR</td>
<td>constrained pressure residual</td>
</tr>
<tr>
<td>AMG</td>
<td>algebraic multi-grid</td>
</tr>
<tr>
<td>MSFV</td>
<td>multiscale finite-volume method</td>
</tr>
<tr>
<td>MsFEM</td>
<td>multiscale finite-element method</td>
</tr>
<tr>
<td>AD-GPRS</td>
<td>automatic differentiation based general purpose research simulator</td>
</tr>
</tbody>
</table>
Appendix A

Asymptotic Limit of Capillary Diffusion Coefficient

Assume the phase mobilities are \( \lambda_w = S^\alpha \) and \( \lambda_n = (1 - S)^\beta \), where \( \alpha \geq 1 \) and \( \beta \geq 1 \).

Based on this assumption,

\[
\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} = \frac{S^\alpha (1 - S)^\beta}{S^\alpha + (1 - S)^\beta}.
\]  

(A-1)

Therefore,

\[
\lim_{S \to 0} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} = \lim_{S \to 0} \frac{S^\alpha (1 - S)^\beta}{S^\alpha + (1 - S)^\beta} = \lim_{S \to 0} S^\alpha = 0.
\]

(A-2)

\[
\lim_{S \to 1} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} = \lim_{S \to 1} \frac{S^\alpha (1 - S)^\beta}{S^\alpha + (1 - S)^\beta} = \lim_{S \to 1} (1 - S)^\beta = 0.
\]

(A-3)
Now, we analyse the asymptotic limit of the capillary diffusion coefficient, i.e., \( \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} \), respectively, for Brooks-Corey model and van Genuchten model.

### A.1 Brooks-Corey model

The Brooks-Corey model for drainage capillary pressure curve can be written as:

\[
p_c = p_e S^{-\frac{1}{\gamma}}, \quad \gamma > 1; 
\]  

while for imbibition capillary pressure curve, the Brooks-Corey gives us

\[
p_c = p_e \left( S^{-\frac{1}{\gamma}} - 1 \right), \quad \gamma > 1. 
\]

For both drainage and imbibition curves, we obtain

\[
\frac{dp_c}{dS} = -\frac{p_e}{\gamma} S^{-\frac{1}{\gamma} - 1}. 
\]

Hence,

\[
\lim_{S \to 0} \frac{dp_c}{dS} = -\infty, 
\]

and

\[
\lim_{S \to 1} \frac{dp_c}{dS} = -\frac{p_e}{\gamma}. 
\]

The capillary diffusion coefficient is

\[
\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = \frac{p_e}{\gamma} S^\alpha (1 - S)^\beta S^{-\frac{1}{\gamma} - 1}. 
\]
On one hand, it is easy to obtain that
\[
\lim_{S \to 1} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = 0.
\] (A-10)

That is, the coefficient is always bounded as \( S \to 1 \), On the other hand,
\[
\lim_{S \to 0} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = \lim_{S \to 0} \frac{-p_e}{\gamma} \frac{S^\alpha (1 - S)^\beta}{S^\alpha + (1 - S)^\beta} S^{-\frac{1}{\gamma} - 1}
\]
\[
= \frac{-p_e}{\gamma} \lim_{S \to 0} \frac{S^\alpha}{S^{\frac{1}{\gamma} + 1}}.
\] (A-11)

We can see that as \( S \) approaches to 0, the asymptotic limit of the capillary diffusion coefficient is bounded if \( \alpha \geq \frac{1}{\gamma} + 1 \). Since \( \gamma > 1 \), \( \frac{1}{\gamma} + 1 < 2 \). If quadratic relative permeability curves are used, i.e., \( \alpha = 2 \), the capillary diffusion coefficient is bounded within \( S \in [0, 1] \). In Fig. A.1, we plot the diffusion coefficient for one example of Brooks-Corey model with \( p_e = 10; \gamma = 4; \) and \( \alpha, \beta = 2 \). It is observed that \( \frac{dp_c}{dS} \) is not bounded as \( S \to 0 \) but \( \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \) is bounded. Since \( \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \) goes to 0 faster than \( \frac{dp_c}{dS} \) goes to infinity, the capillarity diffusion coefficient, i.e., \( \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} \), is still bounded.
Figure A.1: Asymptotic limit of capillary diffusion coefficient for one example of Brooks-Corey model. $p_e = 10; \gamma = 4; \text{ and } \alpha, \beta = 2$. (a) $p_c$ vs. $S$; (b) $\frac{dp_c}{dS}$ vs. $S$; (c) $\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n}$ vs. $S$; (d) Diffusion coefficient $\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS}$ vs. $S$. 
APPENDIX A. ASYMPTOTIC LIMIT OF CAPILLARY DIFFUSION COEFFICIENT

A.2 van Genuchten model

The van Genuchten model is

\[ p_c = p_e \left( S^{-\frac{1}{m}} - 1 \right)^{1-m}, \quad 0 < m < 1. \]  

(A-12)

Hence

\[ \frac{dp_c}{dS} = p_e (1 - m) \left( S^{-\frac{1}{m}} - 1 \right)^{-m} \left( -\frac{1}{m} \right) S^{-\frac{1}{m}-1} \]

\[ = -p_e \frac{1 - m}{m} S^{-\frac{1}{m}-1} \left( S^{-\frac{1}{m}} - 1 \right)^{-m}. \]  

(A-13)

Therefore,

\[ \lim_{S \to 0} \frac{dp_c}{dS} = -p_e \frac{1 - m}{m} \lim_{S \to 0} \left( S^{-\frac{1}{m}} - 1 \right)^{-m} \left( S^{-\frac{1}{m}} \right)^{1+1} \]

\[ = -p_e \frac{1 - m}{m} \lim_{S \to 0} \left( S^{-\frac{1}{m}} - 1 \right)^{-m} \left( S^{-\frac{1}{m}} \right)^{1+1} \]

\[ = -\infty, \]  

(A-14)

and

\[ \lim_{S \to 1} \frac{dp_c}{dS} = -p_e \frac{1 - m}{m} \lim_{S \to 1} \left( S^{-\frac{1}{m}} - 1 \right)^{-m} \left( S^{-\frac{1}{m}} \right)^{1+1} \]

\[ = -\infty. \]  

(A-15)

That is, \( \frac{dp_c}{dS} \) is unbounded for both \( S \to 0 \) and \( S \to 1 \).

The capillary diffusion coefficient is

\[ \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = -p_e \frac{1 - m}{m} S^{-\frac{1}{m}-1} \left( S^{-\frac{1}{m}} - 1 \right)^{-m} \frac{S^\alpha (1 - S)^3}{S^\alpha + (1 - S)^\beta}. \]  

(A-16)
On one hand,

\[
\lim_{S \to 0} \lambda_w \frac{\lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = -p e \frac{1 - m}{m} \lim_{S \to 0} S^{-\frac{1}{m} - 1} \left( S^{-\frac{1}{m} - 1} - 1 \right)^{-m} S^\alpha (1 - S)^\beta \frac{S^\alpha + (1 - S)^\beta}{S^\alpha + (1 - S)^\beta} \\
= -p e \frac{1 - m}{m} \lim_{S \to 0} S^\alpha \left( S^{-\frac{1}{m} - 1} - 1 \right)^{-m} S^{\frac{\alpha + 1}{m + 1}} \\
= -p e \frac{1 - m}{m} \lim_{S \to 0} S^{\alpha} S^{\frac{1}{m} + 1} \\
= -p e \frac{1 - m}{m} \lim_{S \to 0} S^{\alpha} 
\]

(A-17)

Since 0 < m < 1, \(\frac{1}{m} > 1\). The diffusion coefficient is bounded as \(S \to 0\) if \(\alpha \geq \frac{1}{m}\), i.e., \(\alpha m \geq 1\).

On the other hand, to investigation the behavior of the diffusion coefficient as \(S\) approaches to 1, instead, we define \(x = 1 - S\) and investigate asymptotic behavior of the diffusion coefficient as \(x \to 0\). \(x\) is indeed the nonwetting phase saturation. Now the diffusion coefficient can be written as

\[
\lambda_w \lambda_n \frac{dp_c}{\lambda_w + \lambda_n} dS = -p e \frac{1 - m}{m} (1 - x)^{-\frac{1}{m} - 1} \left( (1 - x)^{-\frac{1}{m} - 1} - 1 \right)^{-m} \frac{(1 - x)^\alpha x^\beta}{(1 - x)^\alpha + x^\beta} \\
= -p e \frac{1 - m}{m} \frac{(1 - x)^{\alpha - \frac{1}{m} + \frac{1}{m} x^\beta}}{(1 - x)^\alpha + x^\beta} \left( (1 - x)^{-\frac{1}{m} - 1} - 1 \right)^{-m}. \tag{A-18}
\]

Therefore,

\[
\lim_{x \to 0} \lambda_w \lambda_n \frac{dp_c}{\lambda_w + \lambda_n} dS = -p e \frac{1 - m}{m} \lim_{x \to 0} \frac{x^\beta}{(1 - x)^{-\frac{1}{m} - 1}} \left( (1 - x)^{-\frac{1}{m} - 1} - 1 \right)^{-m} \\
= -p e \frac{1 - m}{m} \lim_{x \to 0} \frac{x^\beta}{(1 + \frac{x}{m} + O(x^2) - 1)^m} \\
= -p e \frac{1 - m}{m} \lim_{x \to 0} \frac{x^\beta}{(\frac{x}{m})^m + h.o.t.}. \tag{A-19}
\]
Since $0 < m < 1$ and $\beta \geq 1$, $\beta > m$. Hence, we obtain that

$$\lim_{S \to 1} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = \lim_{x \to 0} \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS} = 0.$$ (A-20)

That is, the diffusion coefficient is always bounded as $S \to 1$.

From the analysis above we can see that for van Genuchten model, the capillary diffusion coefficient is bounded within $S \in [0, 1]$ if $\alpha m \geq 1$. To illustrate the asymptotic limit of the diffusion coefficient, we plot the coefficient for one example of van Genuchten model in Fig. A.2 with $p_e = 10$; $m = 0.5$; and $\alpha, \beta = 2$. It is observed that $\frac{dp_c}{dS}$ is unbounded for both $S \to 0$ and $S \to 1$. But $\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS}$ is bounded within $S \in [0, 1]$. Note that in this case $\alpha m = 1$. Hence the diffusion coefficient approaches to a non-zero constant as $S \to 0$. It approaches to zero if $\alpha m > 1$. 
Figure A.2: Asymptotic limit of capillary diffusion coefficient for one example of van Genuchten model. $p_e = 10; m = 0.5; \text{and } \alpha, \beta = 2$. (a) $p_c$ vs. $S$; (b) $\frac{dp_c}{dS}$ vs. $S$; (c) $\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n}$ vs. $S$; (d) Diffusion coefficient $\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \frac{dp_c}{dS}$ vs. $S$. 

APPENDIX A. ASYMPTOTIC LIMIT OF CAPILLARY DIFFUSION COEFFICIENT
Appendix B

Integration Averaging vs. Upstream Weighting

For the discretized 1D transport problem, there are several numerical methods to calculate the capillary flux at the cell interface. One of them is the integration average, which evaluates the cell-centered integration \( C(S) \) by Eqn. (3.4) and then applies the central difference on \( C \) to obtain the capillary flux at the cell interface (Eqn. (3.2)). This capillary flux plus the viscous and gravitational fluxes from upstreaming come out the total flux for solving 1D hyperbolic conservation law. Note that we do not solve the pressure system here. The constant total-velocity provides the other constraint for 1D problems. Another numerical method to evaluate capillary flux is the phase-based upstream weighting. Upstream weighting, based on the information of phase-flow directions, is widely used in the community of reservoir simulation. We compare the numerical results for capillary flux obtained by integration averaging and upstreaming weighting and validate the usage of integration averaging for 1D transport problem in Eqn. (3.2).

We report two test cases for capillarity segregation, i.e., starting from non-equilibrium initial stage (left-half domain is wetting phase and right-half domain is nonwetting
phase), the system moves toward equilibrium due to capillary forces. Capillary model is spatially homogeneous (i.e., homogeneous capillarity) in the first case, while the capillary models are distinct for right and left-half domains (i.e., heterogeneous capillarity) in the second case. In both cases, the phase-flow directions for nonwetting and wetting are known and fixed based on physical insight, i.e., wetting phase goes to right and nonwetting phase goes to the left.

Case 1: homogeneous capillarity

This case is capillary segregation in a homogeneous domain. We simulate it by using both integration averaging and upstream weighting. The capillary model and initial condition are displayed in Fig. B.1(a) and B.1(b). By comparing the final solutions in Fig. B.1(c) and B.1(d), it is clear to see that integration averaging and phase-based upstream weighting provide almost identical solutions for this case: for homogeneous capillarity, a uniform saturation profile is obtained at equilibriums.

Furthermore, we plot the evolution of the wetting-phase fluxes from an early stage ($t = t_1$) to a late stage ($t = t_5$) in Fig. B.2. Positive flux represents wetting phase flowing from left to right. According to physical insight, wetting-phase flux should be non-negative and the magnitude of flux decreases as time evolves. Fig. B.2 shows the numerical fluxes obtained from the two methods are very close. In fact, numerical fluxes obtained from upstreaming weighting is slightly more accurate than those obtained from integration average. It is observed that numerical fluxes in Fig. B.2(b) are strictly non-negative while in Fig. B.2(a), though hard to see, wetting-phase flux is slightly negative in left-half domain when $t = t_1$. 
Figure B.1: Numerical results: 1D incompressible two-phase segregation with (homogeneous) capillary forces: (a) capillary model; (b) initial condition; (c) final solution obtained from integration averaging; (d) final solution obtained from upstreaming weighting.
Case 2: heterogeneous capillarity

In the second example, \( p_c = p_c(S) \) models are distinct in the left half and right half of the domain. The capillary models are shown in Fig. B.3(a). The segregation starts from non-equilibrium initial stage (Fig. B.3(b)) and moves towards the final equilibrium stage (Fig. B.3(c) and Fig. B.3(d)). Note that for heterogeneous capillary model, saturation distribution is not uniform at equilibrium. There is a discontinuity in the saturation profile, where \( p_c \) satisfies the continuity condition. Comparison between Fig. B.3(c) and B.3(d) illustrate that the final numerical solutions from integration averaging and upstreaming weight are nearly identical.

Fig. B.4 shows the wetting-phase flux from the two numerical methods are very close. Again, the wetting-phase fluxes from upstreaming are strictly non-negative, and hence they are more accurate.

Based on the numerical results for these two examples, we observed that when the phase-flow directions are known, integration average and upstreaming weighting
Figure B.3: Numerical results: 1D incompressible two-phase segregation with (heterogeneous) capillary forces: (a) capillary model; (b) initial condition; (c) final solution obtained from integration averaging; (d) final solution obtained from upstreaming weighting.
Figure B.4: Wetting-phase fluxes: 1D incompressible two-phase segregation with (heterogeneous) capillary forces. \( t = t_1 \) to \( t = t_5 \) is the evolution of simulation time. (a) integration averaging; (b) upstreaming weighting.

always provide similar numerical results, which are physical meaningful.

For transport problem in 1D domain with constant total-velocity as additional constraint, the phase-flow direction can be determined by saturation and total-velocity in the presence of viscous and buoyancy forces (Eqn. 3.3). In the presence of capillary forces, we cannot determine the phase-flow directions with the knowledge of total-velocity and phase saturation alone. Pressure distribution, which is the solution of flow problem, is needed to calculate phase-flow direction when capillary forces are significant. Hence upstream weighting, which required the information of phase-flow direction, is not applicable to calculate capillary flux for 1D transport problem when flow problem is not solved. On the other hand, as we have observed from numerical examples, without information about the flow directions of the fluid phases, integration averaging provides a good approximation for calculating capillary flux within the framework of finite-volume discretization. Therefore, we use integration averaging to calculate capillary flux in 1D transport in Eqn. 3.2.
Lemma A-1. (Local Descent) Let \( R : \mathbb{D} \rightarrow \mathbb{R}^n \) be a piecewise continuously differentiable mapping with \( \mathbb{D} \subseteq \mathbb{R}^n \). That is, \( \mathbb{D} = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \cdots \cup \mathbb{D}_k \), where \( \mathbb{D}_i \) \( (i = 1, k) \) is open domain and \( R \in C^1(\mathbb{D}_i) \). Define the level set of \( R \) as

\[
T(x) := \frac{1}{2} \| R(x) \|^2 = \frac{1}{2} R(x)^T R(x).
\tag{C-1}
\]

Let \( R(x) \neq 0 \) and \( \triangle x \neq 0 \). Then there exists a \( \mu > 0 \), such that \( T(x + s\triangle x) < T(x) \), \( 0 < s < \mu \).

**Proof.** Define

\[
\phi(s) := T(x + s\triangle x).
\tag{C-2}
\]

Since \( R \in C^1(\mathbb{D}_i) \), \( \phi(s) \) is piecewise continuously differentiable. If \( x \in \mathbb{D}_j \), \( \exists \mu \), s.t.

\[
x + s\triangle x \in \mathbb{D}_j, \forall 0 < s < \mu.
\tag{C-3}
\]
We have

\[ \phi'(0) = (R^T R)^T \triangle x \]
\[ = R^T J \triangle x \]
\[ = R^T (-R) \]
\[ = -2T(x) < 0. \]

Therefore,

\[ T(x + s \triangle x) < T(x), \forall 0 < s < \mu. \]  \hspace{1cm} (C-4)
Appendix D

Summary of 1D Solution

In this appendix, we summarize the 1D solution for gravity segregation and displacement under viscous and gravity forces, with linear or quadratic relative permeability curves, and with mobility ratio $M = 0.1$, $M = 1$, or $M = 10$. For each case, we plot the analytical solution and show that the numerical solutions do converge to the analytical solution as mesh in time and space is refined. The evolution of the numerical solutions, from initial condition to the end of simulation, is also displayed. Moreover, we plot the numerical solutions obtained with various timestep sizes and illustrate the timestep truncation errors. Finally, we compare the nonlinear performance (iterations per timestep) between three nonlinear solvers, i.e., standard Newton, Newton with Appleyard chopping and trust-region Newton.

D.1 Gravity segregation

In the first section, 1D solution of gravity segregation is summarized. In Fig. D.1, we plot the flux functions of gravity segregation, respectively, for linear (Fig. D.1(a)) and quadratic (Fig. D.1(b)) relative permeabilities. For linear relative permeability, $k_{rw} = S$ and $k_{rn} = 1 - S$, while for quadratic relative permeability, $k_{rw} = S^2$ and
$k_{rn} = (1 - S)^2$. For each relative permeability, the flux functions are illustrated for various mobility ratios, i.e., $M = 0.1$, $M = 1$ and $M = 10$. Note that for linear relative permeability, the flux functions are concave and there are no inflection points on the flux functions, while for quadratic relative permeability, there are inflections points on the flux functions where the convexity changes. Also note that the mobility ratio affects the magnitude, as well as, nonlinearity of the flux functions.

![Flux functions of gravity segregation, respectively, for linear and quadratic relative permeabilities.](image)

(a) linear rel perm  
(b) quadratic rel perm

Figure D.1: Flux functions of gravity segregation, respectively, for linear and quadratic relative permeabilities. (a) Linear relative permeability: $k_{rw} = S$ and $k_{rn} = 1 - S$. (b) Quadratic relative permeability: $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. Red dots represent inflection points and green dots are sonic points.

The initial condition is wetting (heavy) phase on top of nonwetting (light) phase, each phase taking up half of the domain. Then the two phases segregated with buoyancy as the only driven force (as shown in Fig. [D.2]).

For the gravity segregation, we plot the characteristic waves in Fig. [D.3] Starting from the initial condition in Fig. [D.2] saturations to the right of the sonic point move upwards, and those to the left of the sonic point move downwards (as shown in Fig. [D.3(a)]. The saturation at the sonic point will not move. Define $S^{\text{shock}1}$ and $S^{\text{shock}2}$, respectively, for wetting phase shock and nonwetting phase shock, are the
APPENDIX D. SUMMARY OF 1D SOLUTION

172

Figure D.2: Initial condition of gravity segregation

saturation at shock fronts. Given the initial saturation distribution as in Fig. [D.2], we see that one saturation at the interface will remain constant and that two shocks will propagate, wetting-phase shock downwards with shock-front saturation $S^{\text{shock}1}$ and nonwetting-phase shock moving upwards with shock-front saturation $S^{\text{shock}2}$. The speeds of these two shocks are $f'(S^{\text{shock}1})$ and $f'(S^{\text{shock}2})$, respectively.

Considering a time $t_{\text{shock}}$, which precedes the arrival of either shock at the boundaries. Once the shocks arrive at top and bottom of the domain, reflection shocks appear (as shown in Fig. [D.3(b)]. As soon as the nonwetting shock arrives at the bottom of the domain, the saturation must go to 0 in order to satisfy $v_n = 0$. Similarly, as soon as the wetting shock arrives at the top of the domain, the saturation is 1 to make sure that $v_w = 0$. This creates a new discontinuity in saturation at the boundaries, and sets of shocks moving in the opposite direction. Now the shock-front saturations are $S^{\text{reflec}1}$ and $S^{\text{reflec}2}$, respectively, with $[S^{\text{reflec}1}, S^{\text{shock}1}]$ and $[S^{\text{shock}2}, S^{\text{reflec}2}]$ travelling as spreading waves, as in Fig. [D.3(b)]. Assuming the length of the domain is $H$, the propagation time for shock waves is

$$t_{\text{shock}} = \frac{H/2}{f'(S^{\text{shock}})},$$

(D-1)
Figure D.3: Shocks in analytical solution of 1D gravity segregation. (a) Shocks. (b) Reflection shocks.
and the propagation time for reflection shock waves is

\[ t_{\text{reflec}} = \frac{H/2}{f'(S_{\text{reflec}})} \]  

(D-2)

Hence, the time for segregation, i.e., the slower reflection shock arrives at \( H/2 \), is

\[ t_{\text{seg}} = \max (t_{\text{shock1}} + t_{\text{reflec2}}, t_{\text{shock2}} + t_{\text{reflec1}}) . \]  

(D-3)

In the following, we present the solution of gravity segregation for both linear and quadratic relative permeability curves, i.e.,

- \( k_{rw} = S; \quad k_{rn} = 1 - S \)
- \( k_{rw} = S^2; \quad k_{rn} = (1 - S)^2 \)

**Linear relative permeability**

First, we study the solution of gravity segregation when the relative permeability is linear. Table D.1 summarizes the propagation time for the waves and the time for fully segregation from analytical solutions. Note that when the relative permeability is linear, there are no shocks. Starting from the sharp interface between two phases in Fig. [D.2] two spreading waves move in opposite directions, wetting moving downwards and nonwetting moving upwards. The speed of the spreading wave is \( \frac{H/2}{f'} \). Hence when the spreading wave arrives at the domain boundaries,

\[ t_{\text{wave}} = \frac{H/2}{f'(0)} . \]  

(D-4)

Once the spreading waves arrive at the boundaries, two new, i.e., reflection, spreading waves appear and start to move from the domain boundaries. The propagation time
for the fastest saturation front on the reflection wave is

\[ t_{\text{reflect}} = \frac{H}{2 f'(1)}. \]  

(D-5)

In Table D.1, it is noticed that the waves move faster when \( M = 10 \) and slower when \( M = 0.1 \). Also, for different values of mobility ratios, the speeds of nonwetting phase waves and wetting-phase reflection waves are always identical. The reason is that in Fig. D.1(a) the slope of the flux functions are the same at \( S = 1 \) when \( M = 0.1, M = 1 \) or \( M = 10 \).

<table>
<thead>
<tr>
<th>( M )</th>
<th>Wetting</th>
<th>Nonwetting</th>
<th>( t_{\text{wave}} )</th>
<th>( t_{\text{reflect}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td></td>
<td></td>
<td>400.0801</td>
<td>40.0000</td>
</tr>
<tr>
<td></td>
<td>wetting</td>
<td></td>
<td>40.0000</td>
<td>400.0801</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>40.0802</td>
<td>40.0000</td>
</tr>
<tr>
<td></td>
<td>wetting</td>
<td></td>
<td>40.0000</td>
<td>40.0802</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>4.0805</td>
<td>40.0000</td>
</tr>
<tr>
<td></td>
<td>nonwetting</td>
<td></td>
<td>40.0000</td>
<td>4.0805</td>
</tr>
</tbody>
</table>

We plot both the analytical solutions and the numerical solutions in Fig. D.4. The solid lines represent the analytical solutions and the dashed lines represent the numerical solution. For the numerical solutions shown in Fig. D.4, we use fine mesh in time and space, i.e., \( dx = 0.1 \) and \( dt = 1 \). We compare the evolution of solutions until the first wave arrives at the boundary.

From the comparison in Fig. D.4 we found that the numerical solutions can match the analytical solutions reasonably well, except the saturation distribution around \( S = 0 \) and \( S = 1 \). The match between the solutions validates our numerical method,
i.e., the numerical solutions converge to the true solutions as mesh refined.

The evolutions of the numerical solutions, respectively, for $M = 0.1$, $M = 1$ and $M = 10$, are illustrated in Fig. D.5. Starting from the interface between the two phases, two waves propagate in opposite directions, hit the domain boundaries and then reflect. When $M = 1$, the speed of the two waves are identical, while for $M \neq 1$, one wave is faster than the other.

We can observe that when mobility ratio is low, the evolution of the solution is slow. The flux functions in Fig. D.1(a) show that when the mobility ratio is low, the slope of flux function is small and hence the propagation of wave is slow. This explains the slow evolution of solution associated with the low mobility ratio.

Fig. D.6 shows that time truncation errors appear when timestep sizes are large. Note that the truncation errors are significant when the mobility ratio is low (i.e., $M = 0.1$). This is also corresponding to the small slope of flux function (Fig. D.1(a)) and slow wave propagation associated with the low mobility ratio.

The nonlinear performance for various nonlinear solvers is summarized in Table D.2. We compared the number of iterations per timestep for standard Newton, Newton with Appleyard chopping and trust-region chopping. From Fig. D.1(a), it is observed that there is no inflection points on the flux functions when the relative permeability is linear. The trust-region Newton degenerates to standard Newton since neither inflection nor unit-flux points exist and no flow reversal occurs during gravity segregation. In Table D.2, for various timestep sizes ($\Delta t$) and for various mobility ratios ($M$), it is observed that number of iterations per timestep for trust-region Newton is identical to that for the standard Newton.

Moreover, from Fig. D.2, we can see that, compared with standard Newton and trust-region Newton, Newton with Appleyard chopping always takes more iterations (especially for large timestep sizes), revealing slower convergence. This is because for Newton with Appleyard chopping, the solution update is restricted based on a
Figure D.4: Analytical solutions of gravity segregation with linear relative permeability curves. Solid lines: analytical solutions; dashed lines: numerical solutions.
Figure D.5: Evolution of solutions of gravity segregation with linear relative permeability curves. Total time = 1000.
Figure D.6: Final solutions of gravity segregation with linear relative permeability curves. Total time = 1000.
heuristic estimation, instead of the nonlinearity of the target problem.

Table D.2: Nonlinear performance (iterations per timestep) of gravity segregation with linear relative permeability curves. Total time = 1000.

<table>
<thead>
<tr>
<th>$\triangle t$</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestep</td>
<td>1000</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>0.97</td>
<td>1.88</td>
<td>10.70</td>
<td>92.46</td>
</tr>
<tr>
<td>Appleyard</td>
<td>2.501</td>
<td>3.77</td>
<td>8.0</td>
<td>26</td>
</tr>
<tr>
<td>trust region</td>
<td>2.501</td>
<td>3.73</td>
<td>9.9</td>
<td>49</td>
</tr>
<tr>
<td>$M = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>1.66</td>
<td>10.10</td>
<td>89.02</td>
<td>913.43</td>
</tr>
<tr>
<td>Appleyard</td>
<td>1.374</td>
<td>1.81</td>
<td>4.1</td>
<td>9</td>
</tr>
<tr>
<td>trust region</td>
<td>1.374</td>
<td>1.91</td>
<td>4.8</td>
<td>11</td>
</tr>
<tr>
<td>$M = 10$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>8.21</td>
<td>79.25</td>
<td>914.34</td>
<td>9134.55</td>
</tr>
<tr>
<td>Appleyard</td>
<td>1.277</td>
<td>1.79</td>
<td>5.7</td>
<td>39</td>
</tr>
<tr>
<td>trust region</td>
<td>1.273</td>
<td>1.98</td>
<td>10.7</td>
<td>148</td>
</tr>
</tbody>
</table>

Quadratic relative permeability

Then we study the solution of gravity segregation with quadratic relative permeability. When the relative permeability is quadratic, there are shocks starting at the interface between two phases, moving in opposite direction, as shown in Fig. D.3 (a). Also, once the shocks arrives at the domain boundaries, reflection shocks start to move from the boundaries. The construction of the shocks and reflection shocks for quadratic relative permeability is summarized in Fig. D.7.

Table D.3 shows the saturation front and propagation time for shocks and reflection shocks. Also, the estimation on segregation time is provided for various mobility ratios. Note that the segregation time provided here is the time when the slower reflection shock arrives at the middle of the domain, i.e., $H/2$. In the last column,
Figure D.7: Shocks to construct analytical solution for flux function with quadratic relative permeability curves. Red dots represent inflection points; green dots are sonic points; purple diamonds are shock points; and purple (up) triangles are reflection shock points.
APPENDIX D. SUMMARY OF 1D SOLUTION

it is clear that the segregation is slower when $M = 0.1$ ($t_{\text{seg}} = 3049.0842$) and faster when $M = 10$ ($t_{\text{seg}} = 304.9084$).

Table D.3: Analytical segregation time for gravity segregation with quadratic relative permeability. Column 3: wetting and nonwetting shocks emanating the interface between two phases; column 4: propagation time for the wetting and nonwetting shocks; column 5: wetting and nonwetting reflection shocks emanating from the domain boundaries; column 6: propagation time for the wetting and nonwetting reflection shocks; and column 7: time for fully segregation.

<table>
<thead>
<tr>
<th>$M$</th>
<th>Wetting</th>
<th>Nonwetting</th>
<th>$t_{\text{shock}}$</th>
<th>$t_{\text{reflec}}$</th>
<th>$t_{\text{seg}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.6990</td>
<td>0.7960</td>
<td>816.1541</td>
<td>970.6863</td>
<td>3049.0842</td>
</tr>
<tr>
<td></td>
<td>0.0298</td>
<td>0.0251</td>
<td>324.8631</td>
<td>223.2930</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.3970</td>
<td>0.6030</td>
<td>144.4290</td>
<td>397.2164</td>
<td>541.6453</td>
</tr>
<tr>
<td></td>
<td>0.1100</td>
<td>0.1100</td>
<td>144.4290</td>
<td>397.2164</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.2040</td>
<td>0.3910</td>
<td>32.4863</td>
<td>223.2930</td>
<td>304.9084</td>
</tr>
<tr>
<td></td>
<td>0.2512</td>
<td>0.2985</td>
<td>81.6154</td>
<td>97.0686</td>
<td></td>
</tr>
</tbody>
</table>

In Fig. D.8, we compare the evolution of the numerical solutions (with refined mesh) with that of the analytical solutions, form the initial stage until the one of shocks (from the analytical solutions) arrives at the domain boundary. It is observed that the numerical solutions match the analytical solutions quite well, revealing that the numerical solutions converge to the physical solutions. The dispersion in numerical solutions is due to the discretization error in time and space. We see that the numerical dispersion triggers the reflection shocks, which do not exist in analytical solutions.

Moreover, when $M = 1$, both phase shocks move at the same speed. On the contrary, when $M \neq 1$, one shock moves faster than the other shock.

In Fig. D.9, we plot the evolution of numerical solutions for $M = 0.1$, $M = 1$ and $M = 10$. When $M = 1$, the solution is always symmetric during evolution, revealing identical wave speeds for two phases. Also, we observe slow convergence associated
Figure D.8: Analytical solutions of gravity segregation with quadratic relative permeability curves. Solid lines: analytical solutions; dashed lines: numerical solutions.
APPENDIX D. SUMMARY OF 1D SOLUTION

with low mobility ratio \((M = 0.1)\) and this is due to the small slope of the flux function (Fig. D.1(b)) and corresponding slow propagation of wave.

We compare the solutions with different timestep sizes in Fig. D.10. Time truncation error is not negligible when the timestep size is large \((\Delta t = 1000)\). Also, Fig. D.10 shows that the truncation error is large when \(M\) is small.

For the quadratic relative permeability curves, there are two inflection points and no unit-flux points on the flux functions (Fig. D.1(b)). The trust-region Newton method applies chopping on the two inflection points. The comparison of nonlinear performance between different nonlinear solvers is summarized in Table D.4. It is noticed that when timestep is large, the standard Newton does not converge. Both Newton with Appleyard chopping and the trust-region Newton can converge for all the timestep sizes we have tested, and the nonlinear performance (number of iterations per timestep) for these two nonlinear solvers is comparable.

Table D.4: Nonlinear performance (iterations per timestep) of gravity segregation with quadratic relative permeability curves. Total time = 1000.

<table>
<thead>
<tr>
<th>(\Delta t)</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>timesteps</td>
<td>1000</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>(M = 0.1)</td>
<td>CFL standard</td>
<td>1.78</td>
<td>2.60</td>
<td>10.29</td>
</tr>
<tr>
<td></td>
<td>Appleyard trust region</td>
<td>2.712</td>
<td>3.88</td>
<td>7.9</td>
</tr>
<tr>
<td>(M = 1)</td>
<td>CFL standard</td>
<td>2.15</td>
<td>5.96</td>
<td>44.68</td>
</tr>
<tr>
<td></td>
<td>Appleyard trust region</td>
<td>2.294</td>
<td>2.94</td>
<td>6.1</td>
</tr>
<tr>
<td>(M = 10)</td>
<td>CFL standard</td>
<td>3.24</td>
<td>19.02</td>
<td>159.46</td>
</tr>
<tr>
<td></td>
<td>Appleyard trust region</td>
<td>2.292</td>
<td>3.37</td>
<td>12.1</td>
</tr>
<tr>
<td></td>
<td>trust region</td>
<td>2.366</td>
<td>3.37</td>
<td>11.9</td>
</tr>
</tbody>
</table>
Figure D.9: Evolution of solutions of gravity segregation with quadratic relative permeability curves. Total time = 1000.
Figure D.10: Final solutions of gravity segregation with quadratic relative permeability curves. Total time = 1000.
D.2 Displacement with viscous and buoyancy forces

In the second section, the 1D solutions for displacement with viscous and buoyancy forces are summarized.

We plot the flux functions for both linear and quadratic relative permeabilities in Fig. D.11. When the relative permeability is linear, the flux functions are concave and there are no inflection points on the flux functions (Fig. D.11(a)). With the presence of viscous and buoyancy forces, flow reversal may occur and there are unit-flux points on the flux functions shown in Fig. D.11(a). On the other hand, when the relative permeability is quadratic, there are both inflection points and unit-flux points on the flux functions (Fig. D.11(b)).

![Figure D.11: Flux functions of viscous and buoyancy forces, respectively, for linear and quadratic relative permeabilities. $N_g = -20$. (a) Linear relative permeability: $k_{rw} = S$ and $k_{rn} = 1 - S$. (b) Quadratic relative permeability: $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. Red dots represent inflection points; orange dots are unit-flux points; and green dots are sonic points.](image)

For displacement problems in 1D domain with both viscous and buoyancy forces, initially nonwetting (light) phase is on top of the wetting (heavy) phase (as shown in Fig. D.12). Wetting phase is injected from the top of the domain, and the bottom of
the domain is a free boundary.

Figure D.12: Initial condition of displacement under viscous and buoyancy forces

**Linear relative permeability**

In the first section, 1D problems with linear relative permeability is summarized.

We plot the evolution of solutions in Fig. D.13. It is observed that the viscous forces are balancing the buoyancy forces during the evolution and the initial sharp interface between two phases is fully smeared at the end of the simulation. Also, we can see that when $M = 1$, the solution evolves symmetrically and the final saturation is uniformly distributed at a value of $S = 0.5$ except for the region around the top and bottom boundaries. When $M = 0.1$, the final solution is $S = 0.82$ while when $M = 10$, the final solution is $S = 0.22$.

In Fig. D.14, we compare the numerical solutions obtained from different timestep sizes. We can see the time truncation is not significant, especially when $M = 1$ and $M = 10$.

Table D.5 summarizes the comparison of nonlinear performance between different
Figure D.13: Evolution of solutions of displacement under viscous and buoyancy forces with linear relative permeability curves. Total time = 1000.
Figure D.14: Final solutions of displacement under viscous and buoyancy forces with linear relative permeability curves. Total time = 1000.
nonlinear solvers. All three nonlinear solvers, standard Newton, Newton with Apple- 
yard chopping and trust-region Newton, can converge for any timestep size since the 
flux functions are concave.

Table D.5: Nonlinear performance (iterations per timestep) of displacement under 
viscous and buoyancy forces with linear relative permeability curves. Total time = 
1000.

<table>
<thead>
<tr>
<th>$\triangle t$</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>timesteps</td>
<td>1000</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>8.23</td>
<td>38.15</td>
<td>367.78</td>
<td>3587.76</td>
</tr>
<tr>
<td>Newton</td>
<td>1.629</td>
<td>2.42</td>
<td>3.8</td>
<td>11</td>
</tr>
<tr>
<td>Appleyard</td>
<td>1.635</td>
<td>2.43</td>
<td>4.1</td>
<td>12</td>
</tr>
<tr>
<td>trust region</td>
<td>1.655</td>
<td>2.45</td>
<td>3.7</td>
<td>10</td>
</tr>
<tr>
<td>$M = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>19.95</td>
<td>124.77</td>
<td>1246.89</td>
<td>12460.94</td>
</tr>
<tr>
<td>Newton</td>
<td>1.255</td>
<td>1.62</td>
<td>3.7</td>
<td>14</td>
</tr>
<tr>
<td>Appleyard</td>
<td>1.258</td>
<td>1.63</td>
<td>3.7</td>
<td>15</td>
</tr>
<tr>
<td>trust region</td>
<td>1.256</td>
<td>1.63</td>
<td>4.0</td>
<td>12</td>
</tr>
<tr>
<td>$M = 10$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFL standard</td>
<td>57.76</td>
<td>409.86</td>
<td>4098.57</td>
<td>40843.77</td>
</tr>
<tr>
<td>Newton</td>
<td>1.148</td>
<td>1.39</td>
<td>2.9</td>
<td>15</td>
</tr>
<tr>
<td>Appleyard</td>
<td>1.151</td>
<td>1.45</td>
<td>3.4</td>
<td>14</td>
</tr>
<tr>
<td>trust region</td>
<td>1.148</td>
<td>1.39</td>
<td>2.9</td>
<td>15</td>
</tr>
</tbody>
</table>

Quadratic relative permeability

In the second section, we investigate the 1D solution with quadratic relative perme-
ability.

Fig. D.15 shows that the evolution of solution. When $M = 1$, the waves of the 
two phases propagate with the same speed and collapse around $S = 0.5$ at the end of 
simulation. The final saturation is uniformly distributed, except at the region around 
the top and bottom boundaries. When $M = 0.1$, the nonwetting phase moves faster 
than the wetting phase and two waves collapse around $S = 0.8$ while when $M = 10,$
the wetting phase moves faster than the nonwetting phase and two waves collapse at \( S = 0.25 \).

Fig. D.16 compares the numerical solutions with different timestep sizes. It is observed the time truncation errors are rather small. The numerical solution with very large timestep size (\( \Delta t = 1000 \)) is close to the solution with small timestep size (\( \Delta t = 1 \)), especially when \( M = 1 \) and \( M = 10 \).

We summarize the nonlinear performance (iterations per timestep) for different nonlinear solvers in Table D.6. It shows that when \( M = 0.1 \) or \( M = 1 \), the standard Newton does not converge for timestep size larger than 10 (\( \Delta t = 10 \) corresponds to CFL=24.31 and CFL=124.69, respectively, for \( M = 0.1 \) and \( M = 1 \)). Also, when \( M = 10 \), the standard Newton does not converge when \( \Delta t \geq 1 \), i.e., CFL\( \geq 41.19 \).

On the other hand, both Newton with Appleyard chopping and trust-region Newton can converge, with similar number of iterations, for all the timestep sizes.

Table D.6: Nonlinear performance (iterations per timestep) of displacement under viscous and buoyancy forces with quadratic relative permeability curves. Total time = 1000.

<table>
<thead>
<tr>
<th>( \Delta t )</th>
<th>1</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta t )</td>
<td>1000</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>( M = 0.1 )</td>
<td>CFL</td>
<td>3.71</td>
<td>24.31</td>
<td>223.42</td>
</tr>
<tr>
<td></td>
<td>standard</td>
<td>1.691</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Appleyard</td>
<td>1.654</td>
<td>2.56</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td>trust region</td>
<td>1.750</td>
<td>2.84</td>
<td>4.5</td>
</tr>
<tr>
<td>( M = 1 )</td>
<td>CFL</td>
<td>13.25</td>
<td>124.69</td>
<td>1226.96</td>
</tr>
<tr>
<td></td>
<td>standard</td>
<td>1.653</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Appleyard</td>
<td>1.253</td>
<td>1.64</td>
<td>4.0</td>
</tr>
<tr>
<td></td>
<td>trust region</td>
<td>1.261</td>
<td>1.66</td>
<td>4.2</td>
</tr>
<tr>
<td>( M = 10 )</td>
<td>CFL</td>
<td>41.19</td>
<td>411.88</td>
<td>4118.78</td>
</tr>
<tr>
<td></td>
<td>standard</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Appleyard</td>
<td>1.140</td>
<td>1.39</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>trust region</td>
<td>1.150</td>
<td>1.40</td>
<td>3.4</td>
</tr>
</tbody>
</table>
Figure D.15: Evolution of solutions of displacement under viscous and buoyancy forces with quadratic relative permeability curves. Total time = 1000.
Figure D.16: Final solutions of displacement under viscous and buoyancy forces with quadratic relative permeability curves. Total time = 1000.
Appendix E

Trust-region Newton vs. Continuation Newton

In this appendix, we compare the nonlinear performance between our trust-region Newton in Chapter 3 and the Continuation Newton method (CN) in Younis [28] and Younis et al. [11]. For the description of the CN, see section 2.3.

Algorithm F.1 prescribes the general CN process to solve a target timestep $\Delta t_{\text{target}}$ given an old state $S^n$, which may involve many unknowns. Algorithm F.1 uses several sub-algorithms. The APPLEYARD-SAFE-UPDATE performs a cell-wise saturation update along the tangent direction, and the NEWTON-CORRECTOR and NEWTON-SOLVER perform a Newton process using the trust-region Newton method. In Algorithm F.1, local Newton corrections to bring points closer to the solution path are performed by a Newton solver, except that a different looser convergence tolerance is used. Specially, the corrector convergence tolerance only needs to be smaller than, or equal to, the convergence neighborhood tolerance. The remaining sub-algorithms are developed as follows: COMPUTE-TANGENT in Algorithm F.2 which computes the normalized tangent update vector, $\delta$, SELECT-TANGENT-STEP-LENGTH in Algorithm F.3 which performs a search along the tangent for the...
APPENDIX E. TRUST-REGION NEWTON VS. CONTINUATION NEWTON

largest step which remains within the convergence neighborhood, and IS-WITHIN-NEIGHBORHOOD in Algorithm [F.3] which tests whether a point is inside the convergence neighborhood.

Algorithm F.1 CONTINUATION-NEWTON-STEP ($S^n, \Delta t_{target}$)

Require: $\Delta t_{target} \geq 0$ and $S^n \in \mathbb{R}^N$

Ensure: $R(S, \Delta t; S^n) = 0, 0 < \Delta t \leq \Delta t_{target},$ and $\nu < N_{max}$

1. $\nu \leftarrow 0$
2. $\Delta t \leftarrow 0$
3. $S \leftarrow S^n$

while $\nu < N_{max},$ and, $\Delta t < \Delta t_{target}$ do

4. $\delta \leftarrow$ COMPUTE-TANGENT($S, \Delta t, S^n$)
5. $\alpha \leftarrow$ SELECT-TANGENT-STEP-LENGTH($S, \Delta t, \delta, S^n$)
6. $W \leftarrow S$
7. for $i = 0$ to $N$ do
8. $W(i) \leftarrow$ APPLEYARD-SAFE-UPDATE($W(i), \alpha \delta(i)$)
9. end for
10. $\Delta t_2 \leftarrow \alpha \delta(N + 1)$
11. $\nu \leftarrow \nu + 1$

12. if IS-WITHIN-NEIGHBORHOOD($W, \Delta t_2, S^n$) then
13. $S \leftarrow W$
14. $\Delta t \leftarrow \Delta t_2$
15. else
16. $S, N_{newton} \leftarrow$ NEWTON-CORRECTOR($S, \Delta t; S^n$)
17. $\nu \leftarrow \nu + N_{newton}$
18. end if
19. end while
20. $S, N_{newton} \leftarrow$ NEWTON-SOLVER($S, \Delta t; S^n$)
21. $\nu \leftarrow \nu + N_{newton}$
22. return $S, \Delta t, \nu$

Then we study the performance of CN for single-cell transport problems. We test two scenarios: 1) with viscous forces only, and 2) with viscous and buoyancy forces. In both cases, $\Delta t_{target} = 150,$ and the cell is fully saturated by non-wetting phase at the beginning. Wetting phase is injected into the cell. We compare the performance of CN with our trust-region Newton method. As will be illustrated in the numerical
Algorithm F.2 COMPUTE-TANGENT \((S, \Delta t, S^n)\)[28]

Require: \(\Delta t \geq 0\) is a timestep size for state \(S \in \mathbb{R}^N\) from the initial state \(S^n \in \mathbb{R}^N\).

Ensure: \(\delta \in \mathbb{R}^{N+1}\) is the unit tangent to the augmented solution curve emanating from \((S^n, 0)\) and \((S^n, \Delta t)\)

Fix a positive constant \(C\).

\[ C \leftarrow \max(C, \frac{1}{\max_{S} \| \frac{\partial R(S; \Delta t, S^n)}{\partial \Delta t} \|} ) \]

\[
\delta_S \leftarrow -C J^{-1} \frac{\partial R(S; \Delta t, S^n)}{\partial \Delta t}
\]

\[
\delta_{\Delta t} \leftarrow C
\]

\[
\delta \leftarrow (\delta_S, \delta_{\Delta t})^T
\]

\[
\delta \leftarrow \frac{1}{\| \delta \|} \delta
\]

return \(\delta\)

Algorithm F.3 IS-WITHIN-NEIGHBORHOOD\((S, \Delta t, S_0)\)[28]

Require: \(\Delta t \geq 0\) is a timestep size for state \(S \in \mathbb{R}^N\) from the initial state \(S_0 \in \mathbb{R}^N\).

Ensure: Returns whether \(S\) is within the convergence neighborhood at timestep \(\Delta t\) from the state \(S_0\)

Fix a positive tolerance \(\epsilon_{\text{neighborhood}}\).

\[ \text{if } \| R(S, \Delta t; S_0) \| < \epsilon_{\text{neighborhood}} \]

\[ \text{return true} \]

else

\[ \text{return false} \]

end if
Algorithm F.4 SELECT-TANGENT-STEP-LENGTH($S, \Delta t, \delta, S^n$)[28]

Fix a maximum number of backtracking iterations; $\nu_{\text{max}} \leq 1$
Fix a minimal timestep size advancement per tangent step, $\Delta t_{\text{min}} > 0$. This can be based on a suitable CFL number such as one.
Fix a maximal timestep size advancement per tangent step, $\Delta t_{\text{max}} > \Delta t_{\text{min}}$. A typical CFL number may be ten.

\[
\begin{align*}
\alpha_{\text{min}} &\leftarrow \frac{\Delta t_{\text{min}}}{\delta(N+1)} \\
\alpha_{\text{max}} &\leftarrow \frac{\Delta t_{\text{max}}}{\delta(N+1)} \\
\Delta &\leftarrow \frac{\alpha_{\text{max}} - \alpha_{\text{min}}}{\nu_{\text{max}}} \\
\alpha &\leftarrow \alpha_{\text{max}} \\
\end{align*}
\]

repeat

for $i = 0$ to $N$ do

$S^*(i) \leftarrow$ APPLEYARD-SAFE-UPDATE($S(i), \alpha \delta(i)$)

end for

$\Delta t^* \leftarrow \Delta t + \alpha \delta(N + 1)$

is-converged $\leftarrow$ IS-WITHIN-NEIGHBORHOOD($S^*, \Delta t^*, S^n$)

$\alpha \leftarrow \alpha - \Delta$

until $\alpha < \alpha_{\text{min}} - \Delta$ or is-converged=true

return $\alpha$


Evidences below, the nonlinear performance of CN is sensitive to some of its heuristic parameters. Specifically, the number of tangent steps and the number of correction iterations in CN method depend on the value of $\alpha_{\text{min}}$ and $\alpha_{\text{max}}$ in Algorithm F.4 as well as, on the selected $\epsilon_{\text{neighborhood}}$ in Algorithm F.3.

For the first numerical example, single-cell transport with viscous forces only, we choose $\epsilon_{\text{neighborhood}} = 8e - 1$. $\alpha_{\text{min}}$ and $\alpha_{\text{max}}$ are chosen, respectively, corresponding to a CFL number of 1 and 10. The solution path (red curve), convergence neighborhood (black dashed curve), intermediate solutions after tangent steps (blue dots) and correction steps (yellow dots) of CN method, and iterative solutions of our trust-region Newton methods (red dots) are displayed in Fig. E.1 where the X axis represents the saturation value and the Y axis represents the timestep size. Note that, each intermediate solution after a correction step in CN (yellow dots in Fig. E.1) corresponds to a solution associated with a timestep size that is smaller than the target timestep.
size, i.e., $\Delta t < \Delta t_{target}$. On the other hand, the iterative solutions of our trust-region Newton method always lie on the top horizontal line ($\Delta t = \Delta t_{target}$) in Fig. E.1. For this case, CN takes 11 tangent steps, and 15 Newton iterations in the correction steps while our trust-region Newton method is able to converge within 8 iterations. Thus, for this case, our trust-region Newton method yields a better convergence behavior than the CN method.

![Figure E.1](image)

Figure E.1: Solution for single-cell transport with viscous forces, respectively, from CN and trust-region Newton methods. $N_g = 0$. Blue dots: tangent step in CN; yellow dots: correction step in CN; red dots: iterative solutions from trust-region Newton; red solid line: solution path; black dashed line: convergence neighborhood in CN.

For the second numerical example, single-cell transport with both viscous and buoyancy forces. $N_g = -5$. We choose $\epsilon_{\text{neighborhood}} = 8\epsilon - 1$. For this case, $\alpha_{\text{min}}$
and $\alpha_{\text{max}}$ are chosen, respectively, corresponding to a CFL number of 1 and 50. The solution path, convergence neighborhood, and intermediate solutions from both CN and the trust-region Newton methods are plotted in Fig. E.2. We can observe in Fig. E.2 that when buoyancy forces are significant, given the same $\epsilon_{\text{neighborhood}}$, the convergence neighborhood is much narrower compared with the previous case where only viscous forces are present (Fig. E.1). This is due to the nature of self-sharpening in buoyancy-dominated transport problems. For this case, CN takes 16 tangent steps, and 1 Newton iteration in the correction steps. Trust-region Newton method is able to converge within 6 iterations. Again, better nonlinear performance is achieved by our trust-region Newton method. Keeping the parameters $\epsilon_{\text{neighborhood}}$ and $\alpha_{\text{min}}$ unchanged, if we choose $\alpha_{\text{max}}$ corresponding to a CFL number of 10 (like how we choose it in the first example), CN will need to take more than 60 tangent steps and 1 Newton correction step to converge. That is, the performance of CN is highly dependent on the parameters in the algorithms while these parameters are heuristic and case-dependent. There are no robust a-priori estimates for these parameters, which can only be tuned on a trial-and-error basis.
Figure E.2: Solution for single-cell transport with viscous and buoyancy forces, respectively, from CN and trust-region Newton methods. $N_g = -5$. Blue dots: tangent step in CN; yellow dots: correction step in CN; red dots: iterative solutions from trust-region Newton; red solid line: solution path; black dashed line: convergence neighborhood in CN.
Appendix F

Other Examples of Preconditioning Strategy

In this appendix, we summarize the performance of the nonlinear preconditioning strategy in Chapter 5 for 1D transport problems, and 2D flow and transport problems with sequential-implicit method.

F.1 1D transport problems

First, we test the preconditioning strategy for 1D transport problems with viscous forces, viscous and gravitational forces, and gravity segregation where only buoyancy forces are present. Constant total velocity is assumed in all the examples of this section.

F.1.1 Viscous forces

For 1D (1000 grid blocks) transport problems in the presence viscous forces, we assume that the total-velocity is constant, and flow is from left to right. The initial condition
is $S = 0$ and wetting phase ($S = 1$) is injected from the left boundary. We test the transport problems for four types of the relative permeability curves:

- $k_{rw} = S^2; k_{rn} = (1 - S)^2$
- $k_{rw} = S^3; k_{rn} = (1 - S)^3$
- $k_{rw} = S^{10}; k_{rn} = (1 - S)^2$
- $k_{rw} = S^{10}; k_{rn} = (1 - S)^{10}$

In Fig. F.1, we plot the flux functions corresponding to each type of relative permeability curves listed above. The figure indicates that flux functions vary significantly with the mobility ratios when $k_{rw} = S^2$, $k_{rn} = (1 - S)^2$, and quite mildly when $k_{rw} = S^{10}$, and $k_{rn} = (1 - S)^{10}$. Compare the flux function for $M = 10$ (unfavorable displacement) with that of $M = 0.1$ (favorable displacement), it is observed that when $M = 10$, and the slope of the flux function is large for $S \in [0, S^{\text{injlec}}]$, the slope is relatively small for $S > S^{\text{injlec}}$. If $S = 0$ is the initial condition and $S = 1$ is the injection condition, a shock forms starting from $S = 0$, followed by a spreading wave to $S = 1$. We can see that when $M = 10$ (unfavorable displacement), the shock speed is relatively high, resulting in early breakthrough, and the spread wave moves slowly. Correspondingly, the solution front will propagate for a longer distance and hence the spread wave also needs to propagate for a longer distance until it reaches the solution front. Therefore, for the same timestep size, we expect more nonlinear iterations needed by the case with $M = 10$ (unfavorable displacement) than that with $M = 0.1$ (favorable displacement).

For the first type of relative permeability curve, $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$, we simulate the 1D transport problem until $T = 100$ CPVI (cell pore-volume injected) with various mobility ratios: $M = 0.1$, 1 and 10. The timestep size is $\Delta t = 100$ CPVI. For this case, we test the nonlinear performance for the standard Newton,
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

Figure F.1: Flux functions for viscous fluxes: (a) $k_{rw} = S^2$, $k_{rn} = (1 - S)^2$; (b) $k_{rw} = S^3$, $k_{rn} = (1 - S)^3$; (c) $k_{rw} = S^{10}$, $k_{rn} = (1 - S)^2$; (d) $k_{rw} = S^{10}$, $k_{rn} = (1 - S)^{10}$. Red dots represent inflection points.
standard Newton with preconditioning, trust-region Newton, and trust-region Newton with preconditioning. The performance for these nonlinear solvers is summarized in Table F.1. It is clear that the standard Newton method (without preconditioning) is not convergent for any mobility ratio. On the other hand, the trust-region Newton (without preconditioning) is convergent taking 154 iterations for $M = 0.1$, 262 iterations for $M = 1$, and 602 iterations for $M = 10$. Although the trust-region Newton method guarantees convergence, the rate of convergence for this case is too low. The table indicates that preconditioning accelerates the convergence for both standard Newton and the trust-region Newton significantly. With preconditioning, whereby the inflection-point saturation is used as an initial guess, both nonlinear solvers can converge in less than 10 iterations, for the three mobility ratios. That is, we can save more than one order of magnitude in the computational cost by using our preconditioning strategy.

Table F.1: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous forces. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. $T = 100$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>292.73</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>200.01</td>
<td>-</td>
<td>262</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>298.85</td>
<td>-</td>
<td>602</td>
</tr>
</tbody>
</table>

It is noticed in Table F.1 that with preconditioning, the numbers of Newton iterations by using standard Newton (the 4th column) is the same as the ones by using the trust-region Newton (the 6th column), for various mobility ratios. This can be explained as follows. In the presence of viscous forces only, the trust-region Newton method is applying chopping at the (unique) inflection point. Once the iterative solution or the initial guess resides in the same side of the inflection point as
the final solution, convergence is guaranteed. With preconditioning, the initial guess is the inflection point, hence the standard Newton method will achieve the same nonlinear performance as the trust-region Newton method, as long as the iterative solution stays within the same trust region as the true solution. This condition is likely to be valid when the iterative solution starts from an initial guess with a small convergence ratio (e.g., the inflection point) and the timestep size is not extremely large. Two Newton methods are essentially identical under such circumstances. The similar phenomena will also be observed in the following 1D transport problems with viscous flux.

To illustrate how preconditioning accelerates the convergence rate, we plot the initial guess, the sequence of iterative updates and the converged solution for the trust-region Newton method in Fig. F.2. The saturation distribution during the iterative process is shown for various mobility ratios ($M = 0.1$, $M = 1$, and $M = 10$), without and with preconditioning strategy.

Without preconditioning, the initial guess for the nonlinear solver is the same as the initial condition, i.e., $S = 0$. Refer to the blue dashed lines in Fig. F.2(a), F.2(c) and F.2(e). We observe that, if no preconditioning is employed, there are ‘spikes’ in the iterative solutions. The ‘spikes’ move slowly as the Newton iterations increase. Hence, the nonlinear solver consumes a significant amount of iterations to converge. On the other hand, using the inflection point as the initial guess for the first Newton iteration (blue dashed lines shown in Fig. F.2(b), F.2(d) and F.2(f)) instead of $S = 0$ leads to fast convergence. Note that for different mobility ratios ($M = 0.1$, $M = 1$ and $M = 10$), the inflection points are different. Hence, the initial guess is different for the different cases. With preconditioning, the iterative solutions are monotonic, and there are no ‘spikes’ in the solution. Starting from the initial guess (blue dashed lines in Fig. F.2(b), F.2(d) and F.2(f)), i.e., the inflection point, the first iterative solution (green solid lines) is monotonic. Mass in every grid block is moving no slower
Figure F.2: Solutions of trust-region Newton method for 1D transport with viscous forces. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. $T = 100$.
than the mass flow of upwind grid block, and there are no ‘spikes’ due to unphysical mass accumulation. During the following iterations, the saturation distribution gets sharpened and converges quickly to the solution.

For the second type of relative permeability curves, $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$, we summarize the nonlinear performance (number of Newton iterations) for both standard Newton and trust-region Newton, without and with preconditioning, in Table F.2. A single timestep size $\Delta t = 100$ is used. The third column in Table F.2 shows that without preconditioning standard Newton does not converge for any of the mobility ratios. And the fifth column shows that the trust-region Newton method is convergent, but that the convergence rate is quite slow. On the other hand, we can see that the preconditioning strategy accelerates the convergence for both the standard Newton and the trust-region Newton quite significantly. With preconditioning, both nonlinear solvers can converge in 9 iterations, for the three mobility ratios.

Table F.2: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous forces. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $T = 100$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>349.49</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>300.01</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>353.31</td>
<td>-</td>
<td>9</td>
</tr>
</tbody>
</table>

For this case, we plot the iterative solutions from the trust-region Newton method in Fig. F.3 without and with preconditioning, for $M = 0.1$, 1 and 10. Comparing Fig. F.3 with Fig. F.2 it is clear that the converged solutions with the cubic relative permeability curves are less dispersed than those with quadratic relative permeability curves.
For different mobility ratios, at the end of simulation: $T = 100$, the converged solution (light blue solid lines) with $M = 10$ propagates a longer distance than that with $M = 1$ or 0.1. This is because when $M = 10$, the shock speeds are high. Correspondingly, we found that when $M = 10$, the trust-region Newton method (without preconditioning) takes more iterations to converge since the ‘spikes’ in the iterative solutions moves a longer distance to reach the converged solution. The wave speed in one grid block is lower than that in its upwind grid block, resulting in unphysical mass accumulation and ‘spikes’ in the saturation distribution. The ‘spikes’ move as slow as one grid block per iteration. The Newton method converges only when the ‘spikes’ reach the solution front. On the contrary, by using preconditioning whereby the inflection point (shown as the blue dashed lines in Fig. F.3(b), F.3(d) and F.3(f)) is the initial guess, the wave speed in every grid block is no slower that in an upwind grid block. There is no unphysical mass accumulation and the iterative solutions are always monotonic. From then on, the Newton iterates converge to the solution rapidly.

The third type of relative permeability curves is $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^2$. We provide different exponents for the wetting and nonwetting relative permeability curves. Table F.3 summarized the nonlinear performance for different nonlinear solvers with various mobility ratios ($M = 0.1$, $M = 1$ and $M = 10$). Similar to the previous cases, we can see that for any mobility ratio, the standard Newton (without preconditioning) is not convergent, and the trust-region Newton method (without preconditioning) converges, but quite slowly taking more than 100 iterations. On the contrary, with preconditioning, both nonlinear solvers converge within 10 iterations.

We plot the iterative solutions from the trust-region Newton method in Fig. F.4. Due to the low shock speeds with $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^2$, the converged solution fronts is less dispersive (sharper) than those with quadratic, or cubic, relative
Figure F.3: Solutions of trust-region Newton method for 1D transport with viscous forces. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $T = 100$. 
Table F.3: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous forces. \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^{10} \). T = 100.

<table>
<thead>
<tr>
<th>( M )</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>( M = 0.1 )</td>
<td>646.16</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>( M = 1 )</td>
<td>537.44</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>( M = 10 )</td>
<td>531.86</td>
<td>-</td>
<td>10</td>
</tr>
</tbody>
</table>

permeability curves. Without preconditioning, the iterative solutions are not monotonic, and hence the convergence is slow. With preconditioning, the initial guess is the inflection point for every grid block. Note that having different exponents in the relative permeability curves for the two phases, the inflection points for the various mobility ratios are all above \( S = 0.5 \). Starting from the inflection point, the sequence of iterative updates are always monotonic and converge to the solution quickly.

The fourth type of relative permeability curves for 1D transport problem in the presence of viscous forces only: \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^{10} \). We summarize the performance for nonlinear solvers for this type of relative permeability curves in Table F.4. The standard Newton (without preconditioning) does not converge for any mobility ratio. And the trust-region Newton method (without preconditioning) converges slowly. More iterations are required for \( M = 10 \) than for \( M = 0.1 \) and 1 due to the high shock speed with \( M = 10 \). On the contrary, if preconditioning is applied, both the standard Newton and the trust-region Newton methods converge in 10 iterations.

Fig. F.5 illustrates the iterative solutions with \( k_{rw} = S^{10} \) and \( k_{rn} = (1 - S)^{10} \). The difference in the solution for different mobility ratios (\( M = 0.1, 1 \) and 10) is small, since the flux functions with different mobility ratios are quite similar to each other (Fig. F.1(d)). We observed ‘spikes’ unphysical mass accumulation in the iterative
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

Figure F.4: Solutions of trust-region Newton method for 1D transport with viscous forces. $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^2$. $T = 100$. 

(a) $M=0.1$, no precondition

(b) $M=0.1$, with precondition

(c) $M=1$, no precondition

(d) $M=1$, with precondition

(e) $M=10$, no precondition

(f) $M=10$, with precondition
Table F.4: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous forces. $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^{10}$. $T = 100$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>1012.02</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>1000.01</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>1014.61</td>
<td>-</td>
<td>9</td>
</tr>
</tbody>
</table>

solutions with the trust-region method (Fig. F.5(a), F.5(c) and F.5(e)), which is the reason for the slow convergence. The ‘spikes’ do not exist in the iterative solutions for Newton with preconditioning. That is, the iterative solutions are always monotonic in a way that the wave velocity at any cell is no slower than its upwind cell. Hence, the nonlinear solver with the preconditioning strategy can converge much more quickly than that without preconditioning.

### F.1.2 Viscous and gravitational forces

Now we discuss 1D transport in the presence of viscous and strong buoyancy forces, i.e., $N_g = -5$. The initial condition is $S = 0$ and the wetting phase ($S = 1$) is injected from the left boundary. The direction of gravity is the same as the direction of total velocity, i.e., from left to right. That is, buoyancy helps the injected wetting phase move from left to right. We test the numerical examples with two different types of relative-permeability curves:

- $k_{rw} = S^2; k_{rn} = (1 - S)^2$
- $k_{rw} = S^3; k_{rn} = (1 - S)^3$

For each type of the relative permeabilities curves, we plot the flux functions for $M = 0.1, 1, and 10$ in Fig. F.6. It is observed that flux functions for unfavorable
Figure F.5: Solutions of trust-region Newton method for 1D transport with viscous forces. $k_{rw} = S^{10}$ and $k_{rn} = (1 - S)^{10}$. $T = 100$. 
displacement \((M = 10)\) have larger slope than those for favorable displacement \((M = 0.1)\). Note that when \(M = 0.1\), the flux functions in both Fig. F.6(a) and Fig. F.6(b) are monotonically increasing, and there are no unit-flux points. Also, we can see that for both quadratic and cubic relative permeability curves, the slopes of the flux functions are zero at \(S = 0\) and \(1\), for \(M = 0.1\), \(1\) and \(10\). That is, if the initial guess is \(S = 0\), the low speed of wave propagation and the resulting slow convergence of the nonlinear solvers cannot be avoided.

First, for the quadratic relative permeability curves \((k_{rw} = S^2, k_{rn} = (1 - S)^2)\), we simulate the 1D transport problem under viscous and buoyancy forces using both standard Newton and the trust-region Newton methods. At the end of the simulations: \(T = 100\). And the timestep size is \(\Delta t = 100\). The performance (number of iterations) for \(M = 0.1\), \(1\) and \(10\) is summarized in Table F.5. It is clear that the standard Newton (without preconditioning) does not converge for any mobility ratio. On the contrary, the trust-region Newton method (without preconditioning) converges, but quite slowly. Note that the trust-region Newton takes much more iterations to
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

converge for $M = 10$ than that for $M = 0.1$ (1004 versus 191). In fact, when $M = 10$, $\Delta t = 100$ corresponds to CFL=1121.34, while when $M = 0.1$, $\Delta t = 100$ corresponds to CFL=231.79. This is due to the high shock speed for $M = 10$ compared with that for $M = 0.1$. With preconditioning, both the standard Newton and the trust-region Newton methods converge in seven iterations.

Table F.5: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous and buoyancy forces. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. $N_y = -5$. $T = 100$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>231.79</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>332.40</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>1121.34</td>
<td>-</td>
<td>7</td>
</tr>
</tbody>
</table>

For the 1D two-phase transport problems, we compare the initial guess and the sequence of Newton updates for the trust-region Newton, both with and without preconditioning, in Fig. F.7. Without preconditioning, the initial guess is the initial condition, i.e., $S = 0$, while with preconditioning, the initial guess is $S^{\text{inflce1}} \in (0, S_f=1)$ on the corresponding flux function. We choose $S^{\text{inflce1}} \in (0, S_f=1)$, instead of $S^{\text{inflce2}} \in (S_f=1, 1)$, as the initial guess, since the displacement in 1D domain is an imbibition process. Comparing Fig. F.7 with Fig. F.2 (no gravity), we can see that gravity helps the imbibition process, that is, more wetting fluids invades the 1D domain. Note that when $M = 10$ (Fig. F.7(e) and F.7(f)), breakthrough has already occurred in the final solution.

In Fig. F.7, we observe ‘spikes’ in the sequence of Newton updates without preconditioning. For the grid block that is about to be invaded by the wetting phase, the speed of the wetting phase is zero since $\frac{df}{dS} = 0$ at $S = 0$. On the other hand, its immediate upwind grid block has just been invaded by the injected wetting phase.
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

The speed of the wetting phase prorogation (i.e., $\frac{df}{dS}$ evaluated at the shock point) is much larger than zero. The fluid in the upwind grid block chases the fluid in the downwind grid block, resulting in unphysical mass accumulation. The ‘spikes’ move as slow as one grid block per Newton iteration and cause the slow convergence.

On the contrary, using the preconditioning strategy, the injected wetting-phase fluid in every grid block moves no slower than the fluid in its upwind grid block. Hence, there is no unphysical mass accumulation after the first iteration. The first iterative solution is monotonic throughout the domains (Fig. F.7(b), F.7(d) and F.7(f)). The following iterative solutions are also monotonic, and the Newton methods converge quickly.

Now, we investigate the 1D transport problem under viscous and buoyancy forces with cubic relative permeabilities curves, i.e., $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. The nonlinear performance for this case is summarized in Table F.6. We can see that without preconditioning, standard Newton does not converge and the trust-region Newton method converges quite slowly. With preconditioning, both nonlinear solvers converge quickly. Given that the computation cost per iteration is comparable, speed-ups of 20 to 70 times are expected when the preconditioning strategy is used.

Table F.6: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D transport under viscous and buoyancy forces. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $N_g = -5$. $T = 100$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>324.53</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>350.78</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>755.16</td>
<td>-</td>
<td>10</td>
</tr>
</tbody>
</table>

We plot the sequence of Newton updates from the trust-region Newton method, without and with preconditioning in Fig. F.8. A comparison of Fig. F.8 and Fig.
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

Figure F.7: Solutions of trust-region Newton method for 1D transport with viscous and buoyancy forces. \( N_g = -5 \). \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). \( T = 100 \).
indicates that the solutions with cubic relative permeability curves are less dispersive than those with quadratic relative permeability curves (Fig. [F.6(b)]). We observe ‘spikes’ in solutions from the trust-region Newton without preconditioning. The ‘spikes’ slow the convergence of the nonlinear solver. On the other hand, the iterative solutions from the trust-region Newton with preconditioning are monotonic and converge quickly.

F.1.3 Gravity segregation

Now we investigate the two-phase gravity segregation problem in 1D. The initial condition is $S = 1$ in the top half of the domain, and $S = 0$ in the bottom half of the domain. The boundaries are closed. Both quadratic and cubic relative permeability curves are studied, namely,

- $k_{rw} = S^2; k_{rn} = (1 - S)^2$
- $k_{rw} = S^3; k_{rn} = (1 - S)^3$

We plot the flux functions for the gravity segregation problems in Fig. [F.9]. Notice that for gravity segregation problems, there are no unit-flux points. There is one sonic point and two inflection points. Here, the nonwetting phase displaces the wetting phase (drainage process) in the top half of the domain, whereas the wetting phase displaces the nonwetting phase (imbibition process) in the bottom half of the domain.

For our preconditioning strategy, we use $S^{\text{inflex}1} \in (0, S^f=1)$ for the imbibition region, i.e., the bottom half of the domain, and $S^{\text{inflex}2} \in (S^f=1, 1)$ for the drainage region, i.e., the top half of the domain.

By comparing Fig. [F.9(a)] and Fig. [F.9(b)], it is clear that the slopes of the flux functions are small around $S = 0$ and $S = 1$ in Fig. [F.9(b)] compared with those in Fig. [F.9(a)]. Due to the small slopes and the resulting low wave speeds, segregation problems with cubic relative permeability curves are more challenging for the
Figure F.8: Solutions of trust-region Newton method for 1D transport with viscous and buoyancy forces. $N_g = -5$. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $T = 100$. 
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

nonlinear solvers than the quadratic relative permeability curves.

\[ k_{rw} = S^2, \quad k_{rn} = (1 - S)^2 \]

(a) \( k_{rw} = S^2, \quad k_{rn} = (1 - S)^2 \)

\[ k_{rw} = S^3, \quad k_{rn} = (1 - S)^3 \]

(b) \( k_{rw} = S^3, \quad k_{rn} = (1 - S)^3 \)

Figure F.9: Flux functions for gravity segregation. (a) \( k_{rw} = S^2, \quad k_{rn} = (1 - S)^2 \); (b) \( k_{rw} = S^3, \quad k_{rn} = (1 - S)^3 \). Red dots represent inflection points; and green dots are sonic points.

First, we study gravity segregation with quadratic permeability curves, i.e., \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). A single timestep size \( \Delta t = 200 \) is used. We summarize the nonlinear performance (number of total Newton iterations) in Table F.7 for both standard Newton and trust-region Newton, without and with the preconditioning strategy. Without preconditioning, the standard Newton does not converge for any mobility ratio, and the trust-region Newton is convergent, but quite slowly. We notice that when \( M = 10 \), \( \Delta t = 200 \) corresponds to CFL=351.65, while when \( M = 0.1 \), \( \Delta t = 200 \) corresponds to CFL=18.70. When \( M = 10 \), the trust-region Newton takes much more iterations to converge than that when \( M = 0.1 \) (519 vs. 73).

Table F.7 shows that the preconditioning strategy improves the convergence for both standard Newton and trust-region Newton substantially. With preconditioning, we save more than one order of magnitude in the number of iterations for the trust-region Newton. It is interesting to note that with preconditioning, the standard Newton converges quickly, except for \( M = 10 \). This is explained by analysing the
plots in Fig. F.10.

Table F.7: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D gravity segregation. \( k_{rw} = S^2 \) and \( k_{rn} = (1 - S)^2 \). \( T = 200 \).

<table>
<thead>
<tr>
<th>( M )</th>
<th>CFL</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
</tr>
<tr>
<td>( M = 0.1 )</td>
<td>18.70</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td>( M = 1 )</td>
<td>87.68</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>( M = 10 )</td>
<td>351.65</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

In Fig. F.10, we plot the solutions from trust-region Newton method, without and with preconditioning, for the quadratic relative permeability curves. Starting from the sharp interface between the two phases, i.e., the wetting (heavier) phase moving downwards and the nonwetting (lighter) phase moving upward. There are two shocks moving in opposite directions.

We observed that without preconditioning, there are ‘spikes’ in the sequence of Newton updates. The ‘spikes’ reside in both the top-half (drainage region) and the bottom-half (imbibition region). The ‘spikes’ slow down the Newton convergence substantially. On the other hand, for our preconditioning strategy, \( S^{inflec2} \) is the initial guess for the top-half, and \( S^{inflec1} \) is the initial guess for the bottom-half. By applying the preconditioning strategy, the sequence of Newton solutions are monotonic, and the trust-region Newton converges to the solution quickly.

When \( M = 10 \), the wetting-phase shock moves faster than the nonwetting-phase shock. At the end of simulation, due to the fast wave propagation, the wetting-phase shock reaches the bottom boundary. A reflective shock wave forms and starts to move from the domain boundary (Fig. F.10(e) and F.10(f)). Information about the reflected shock is not captured by the proposed preconditioning strategy. This poses challenges for nonlinear solvers. The preconditioning strategy itself, which is used to
accelerate the convergence, does not guarantee convergence. This explains the non-convergence of standard Newton with preconditioning when $M = 10$, as shown in the fourth column of Table F.7.

Next, we study the gravity segregation with cubic relative permeability curves, i.e., $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. A single timestep $\Delta t = 200$ is used. The number of iterations for both the standard Newton and the trust-region Newton are listed in Table F.8. It is observed that without preconditioning, the standard Newton is not convergent and the trust-region Newton converges slowly for any mobility ratio. We can see that with preconditioning, both the standard Newton and the trust-region Newton methods are convergent, for the three mobility ratios. Due to slow wave propagation with cubic relative permeability curves, all shock waves have not reached the domain boundaries at the end of the simulation.

Table F.8: Performance comparison (number of iterations) between different nonlinear solvers, with and without preconditioning, for 1D gravity segregation. $k_{rw} = S^3$ and $k_{rn} = (1 - S)^3$. $T = 200$.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$\text{CFL}$</th>
<th>standard Newton</th>
<th>trust region</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no precond.</td>
<td>with precond.</td>
<td>no precond.</td>
</tr>
<tr>
<td>$M = 0.1$</td>
<td>15.99</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td>$M = 1$</td>
<td>60.17</td>
<td>-</td>
<td>7</td>
</tr>
<tr>
<td>$M = 10$</td>
<td>202.59</td>
<td>-</td>
<td>8</td>
</tr>
</tbody>
</table>

For this case, the sequence of Newton updates from the trust-region Newton method are displayed in Fig. F.11. Comparing Fig. F.11 with Fig. F.10, we can see that shock waves with the cubic relative permeability curves move slower than those with the quadratic relative permeability curves. This is due to the smaller slope of flux functions with cubic relative permeability curves (as shown in Fig. F.9(b)).

We found that there are ‘spikes’ in the sequence of Newton solutions if preconditioning is not applied while the solutions are monotonic with preconditioning. Note
Figure F.10: Solutions of trust-region Newton method for 1D gravity segregation. $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. $T = 200$. 

$\text{APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY 224}$

\begin{align*}
\text{k}_{rw} &= S^2 \\
\text{k}_{rn} &= (1 - S)^2
\end{align*}
that for our preconditioning strategy, we use different inflection points, as initial
guesses for the top-half domain and the bottom-half domain since these gravity seg-
regation problems, the top-half domain is drainage region and the bottom-half domain
is imbibition region. By applying this preconditioning strategy, the iterative solutions
from the trust-region Newton methods are monotonic and converge to the solution
quickly.
Figure F.11: Solutions of trust-region Newton method for 1D gravity segregation. $k_{ru} = S^3$ and $k_{rn} = (1 - S)^3$. $T = 200.$
F.2 2D flow and transport problems using sequential-implicit method

We now demonstrate our preconditioning strategy of the first Newton iteration for transport in two-dimensional domains with the sequential-implicit method, where the flow (pressure) and transport (saturation) are treated separately and solved in a sequential way. There is a nonlinear flow (pressure) loop followed by a nonlinear transport (saturation) loop. An outer loop is used monitor the error - usually residual tolerance. For the preconditioning strategy used with the sequential-implicit method, see section 5.3.

F.2.1 SPE 10 top layer

First, top layer of the SPE 10 model is used here. It is heterogeneous reservoir containing $60 \times 220 = 13,200$ grid blocks, (see Fig. F.12). The transport is a two-phase, incompressible, wetting-phase flooding problem with relative permeability curves: $k_{rw} = S^2$ and $k_{rn} = (1 - S)^2$. Initially, the reservoir is fully saturated with the non-wetting phase ($S = 0$). We inject the wetting phase at the top-left corner, (1, 1), and produce both wetting and nonwetting phases at the bottom-right corner, (60, 220). We set the same total rate control for both the injector and the producer. We will compare the nonlinear performance between the trust-region Newton method and the trust-region Newton method with preconditioning.

We simulate the 2D transport problem for $T = 10$ days (2.13% PVI). The timestep size is $\Delta t = 10$ days, which corresponds to CFL=139.14. The final saturation distribution is shown in Fig. F.13. At the end of the simulation, a limited region around the injector is swept by the injected wetting phase.

For this case, we compare the nonlinear performance with the sequential-implicit
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY 228

Figure F.12: Permeability of SPE 10 top layer

Figure F.13: Solution of saturation for transport in SPE 10 top layer (60 × 220 = 13,200). End of simulation: $T = 10$ days (2.13% PVI). Timestep size: $\Delta t = 10$ days (CFL=139.14).
method in Table F.9. We list the number of outer iterations (‘outer iter’), the total number of flow (pressure) iterations (‘p iter’), the total number of transport (saturation) iterations (‘S iter’), and the number of transport (saturation) iterations in the first outer iteration (‘S iter within first iter’), between trust-region Newton method and trust-region Newton method with preconditioning. In Table F.9, we can see that the number of outer iterations and the total number of flow (pressure) iterations are the same for the two nonlinear solvers. This shows that preconditioning does not affect the flow (pressure) solvers or the convergence of the outer loop in the sequential-implicit method. On the other hand, it is clear that with preconditioning, the transport (saturation) solver converges much faster, i.e., takes less number of iterations (48 vs. 98), than the trust-region method without preconditioning. In fact, the difference in the total number of transport iterations comes from the difference in the number of transport iterations in the first outer iteration (7 vs. 57). The reason is that the wave velocities are very low at the initial condition and a lot of iterations are needed for the wave to propagate during the first outer iteration, if the initial condition is used as the initial guess. For the rest of the outer iterations, since the transport loop has converged within the previous outer iteration(s), the fluid in every grid block moves at a physical speed, i.e., fluid in one grid block moves no slower than that in its upwind grid block. Thus, there is no need to apply preconditioning for the transport loop in the outer iteration other than during the first iteration.

Now we simulate the same reservoir model with the same initial and boundary conditions until a longer period, i.e., $T = 100$ days (21.3% PVI). The timestep size is changed to $\Delta t = 100$ days, which corresponds to CFL=713.35. The saturation solution at the end of the simulation is shown in Fig. F.14. A large portion of the reservoir is swept by the injected wetting phase.

We summarize the nonlinear performance of the sequential-implicit method in
Table F.9: Performance comparison for transport in SPE 10 top layer (60 × 220 = 13,200). End of simulation: \( T = 10 \) days (2.13% PVI). Timestep size: \( \Delta t = 10 \) days (CFL=139.14). ‘outer iter’ is the number of outer iterations; ‘p iter’ is the total number of flow (pressure) iterations; ‘S iter’ is the total number of transport (saturation) iterations; and ‘S iter within 1st iter’ is the number of transport (saturation) iterations in the first outer iteration.

<table>
<thead>
<tr>
<th></th>
<th>outer iter</th>
<th>p iter</th>
<th>S iter</th>
<th>S iter within first iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust-region</td>
<td>14</td>
<td>44</td>
<td>98</td>
<td>57</td>
</tr>
<tr>
<td>trust-region with precond.</td>
<td>14</td>
<td>44</td>
<td>48</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure F.14: Solution of saturation for transport in SPE 10 top layer (60 × 220 = 13,200). End of simulation: \( T = 100 \) days (21.3% PVI). Timestep size: \( \Delta t = 100 \) days (CFL=713.35).
Table F.10 We report the number of outer iterations, the total number of flow iterations, total number of transport iterations, and the number of transport iterations in the first outer iteration, for the trust-region Newton method, without and with preconditioning. It is observed that there is no difference in the number of outer iterations or the total number of flow iterations between the trust-region Newton method and the trust-region Newton method with preconditioning. The preconditioning strategy only affects the convergence for the transport nonlinear loop in the sequential-implicit method. From the fourth column in Table F.10, we can see the preconditioning strategy accelerates the convergence for transport loop significantly (105 vs. 320). The difference in the total number of transport iterations comes from the difference in the number of transport iterations in the first outer iteration (8 vs. 223), as shown in the fifth column in Table F.10. The difference in the number of iterations is even more significant than that in Table F.9 (shorter period). The reason is that a larger portion of the reservoir has been swept. If we start from the initial condition as the initial guess for simulation, then due to the slow propagation of the transport wave, more iterations are required to propagate the wave before convergence (wave propagates one grid block per iteration) is finally reached. Then, for the following outer iterations, once the transport problem has converged within the first outer iteration, the waves have propagated and arrived at the solution front. Thus, the transport inner loop can converge without difficulties.

F.2.2 SPE 10 bottom layer

Now, we test a two-phase displacement problem using the bottom layer of SPE 10 model (60 × 220 = 13,200 grid blocks). The permeability field is shown in Fig. F.15. Comparing Fig. F.15 and Fig. F.12 (top layer), we can see that the permeability distribution of the bottom layer of SPE 10 model is highly channelized. Fluids in the channels move much faster than those in the region around the channels. The
Table F.10: Performance comparison for transport in SPE 10 top layer \((60 \times 220 = 13,200)\). End of simulation: \(T = 100\) days \((21.3\% \text{ PVI})\). Timestep size: \(\Delta t = 100\) days \((\text{CFL}=713.35)\). ‘outer iter’ is the number of outer iterations; ‘p iter’ is the total number of flow (pressure) iterations; ‘S iter’ is the total number of transport (saturation) iterations; and ‘S iter within 1st iter’ is the number of transport (saturation) iterations in the first outer iteration.

<table>
<thead>
<tr>
<th></th>
<th>outer iter</th>
<th>p iter</th>
<th>S iter</th>
<th>S iter within first iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust-region</td>
<td>29</td>
<td>96</td>
<td>320</td>
<td>223</td>
</tr>
<tr>
<td>trust-region with precond.</td>
<td>29</td>
<td>96</td>
<td>105</td>
<td>8</td>
</tr>
</tbody>
</table>

The contrast of flow velocities introduces challenges for the nonlinear solvers.

The same initial condition and well setting as for the top-layer displacement problems are used, i.e., the reservoir is initially saturated by the nonwetting phase, wetting phase is injected at one corner of the reservoir \((1,1)\), and two phases are produced from the opposite corner \((60,220)\), and we use the quadratic permeability curves: \(k_{rw} = S^2\) and \(k_{rn} = (1 - S)^2\).

First, we simulate the water-flooding problem until 10 days \((2.13\% \text{ PVI})\). Using the timestep size \(\Delta t = 10\) days \((\text{CFL}=147.65)\), the final solution at end of simulation
is shown in Fig. F.16. The sequential-implicit method is used. The nonlinear performance (number of iterations) for trust-region Newton method and the trust-region Newton with preconditioning is compared in Table F.11.

Figure F.16: Solution of saturation for transport in SPE 10 bottom layer ($60 \times 220 = 13,200$). End of simulation: $T = 10$ days (2.13% PVI). Timestep size: $\Delta t = 10$ days (CFL=147.65).

Table F.11 shows that for the sequential-implicit method, the preconditioning strategy again does not affect the outer loop or the flow (pressure) nonlinear loop, i.e., the number of iterations (‘outer iter’) and the total number of flow iterations (‘p iter’) are the same with and without preconditioning. The preconditioning strategy does impact the transport (saturation) loop: the total number of transport iterations (‘S iter’) is greatly reduced, from 91 to 41, with preconditioning. That is, the convergence of transport loop is accelerated by the preconditioning strategy. The reduction in the total number of transport iterations comes mainly from a reduction in the number of transport iterations in the first outer iteration (reduction from 63 to 7). This is due to the fact that for sequential-implicit method, preconditioning is applied once every timestep for the transport loop in the first outer iteration.

We also simulate the bottom-layer displacement problem for a longer period, i.e., $T = 100$ days, corresponding to 21.3% PVI. The timestep size is $\Delta t = 100$ days.
APPENDIX F. OTHER EXAMPLES OF PRECONDITIONING STRATEGY

Table F.11: Performance comparison for transport in SPE 10 bottom layer (60×220 = 13, 200). End of simulation: T = 10 days (2.13% PVI). Timestep size: Δt = 10 days (CFL=147.65). ‘outer iter’ is the number of outer iterations; ‘p iter’ is the total number of flow (pressure) iterations; ‘S iter’ is the total number of transport (saturation) iterations; and ‘S iter within 1st iter’ is the number of transport (saturation) iterations in the first outer iteration.

<table>
<thead>
<tr>
<th></th>
<th>outer iter</th>
<th>p iter</th>
<th>S iter</th>
<th>S iter within first iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust-region</td>
<td>9</td>
<td>28</td>
<td>91</td>
<td>63</td>
</tr>
<tr>
<td>trust-region</td>
<td>9</td>
<td>28</td>
<td>41</td>
<td>7</td>
</tr>
<tr>
<td>with precond.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(CFL=2130.85), and the saturation distribution at the end of the simulation is plotted in Fig. F.17. The injected wetting phase moves along the channels. Comparing Fig. F.17 with Fig. F.16 (shorter period injection), we can see that more wetting phase fluid is propagates deeply into the reservoir.

Figure F.17: Solution of saturation for transport in SPE 10 bottom layer (60×220 = 13, 200). End of simulation: T = 100 days (21.3% PVI). Timestep size: Δt = 100 days (CFL=2130.85).

We compare the number of iterations between the trust-region Newton method and the trust-region Newton with preconditioning in Table F.12. It is clear that preconditioning does not change the number of outer iterations (‘outer iter’) or the
total number of flow (pressure) iterations (‘p iter’), but it reduces the total number of transport (saturation) iterations (‘S iter’) significantly, i.e., from 315 to 65. And this reduction comes mainly from the decreased number of transport iterations in the first outer iteration, i.e., from 263 to 8. Comparing this reduction with that in Table F.11 (shorter period injection), we can see that the reduction for the longer period is more significant than that for the shorter period. With preconditioning, the convergence of the transport nonlinear loop is not limited by the slow wave speed; hence, the transport loop takes a similar number of iterations to resolve the nonlinearity of the transport problems. Without preconditioning, the number of iterations needed by the transport loop is related to the propagation distance of the saturation front. If the simulation ends at a longer period injection, the injected wetting-phase fluid propagates a longer distance. Thus, without preconditioning, the transport inner loop for a longer period takes more iterations to converge than that for a shorter period. That explains the larger reduction in Table F.12 compared with that in Table F.11.

Table F.12: Performance comparison for transport in SPE 10 bottom layer (60 × 220 = 13,200). End of simulation: \( T = 100 \) days (21.3% PVI). Timestep size: \( \Delta t = 100 \) days (CFL=2130.85). ‘outer iter’ is the number of outer iterations; ‘p iter’ is the total number of flow (pressure) iterations; ‘S iter’ is the total number of transport (saturation) iterations; and ‘S iter within 1st iter’ is the number of transport (saturation) iterations in the first outer iteration.

<table>
<thead>
<tr>
<th></th>
<th>outer iter</th>
<th>p iter</th>
<th>S iter</th>
<th>S iter within first iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust-region</td>
<td>14</td>
<td>45</td>
<td>315</td>
<td>263</td>
</tr>
<tr>
<td>trust-region with precond.</td>
<td>14</td>
<td>45</td>
<td>65</td>
<td>8</td>
</tr>
</tbody>
</table>
Bibliography


