An adaptive space-time discontinuous Galerkin method for reservoir flows

by

Yashod Savithru Jayasinghe

M.Eng., University of Cambridge (2013)
S.M., Massachusetts Institute of Technology (2015)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2018

Author . . . .
Department of Aeronautics and Astronautics
May 24, 2018

Certified by . . . .
Thesis Supervisor: David L. Darmofal
Professor of Aeronautics and Astronautics

Certified by . . . .
Thesis Committee Member: Steven R. Allmaras
Research Engineer of Aeronautics and Astronautics

Certified by . . . .
Thesis Committee Member: Ali H. Dogru
Saudi Aramco Fellow

Certified by . . . .
Thesis Committee Member: Youssef M. Marzouk
Professor of Aeronautics and Astronautics

Accepted by . . . .
Hamsa Balakrishnan
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Accepted by . . . .
Nicolas Hadjiconstantinou
Professor of Mechanical Engineering
Co-director, Computational Science and Engineering

An adaptive space-time discontinuous Galerkin method for reservoir flows

by

Yashod Savithru Jayasinghe

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Science and Engineering

Abstract

Numerical simulation has become a vital tool for predicting engineering quantities of interest in reservoir flows. However, the general lack of autonomy and reliability prevents most numerical methods from being used to their full potential in engineering analysis. This thesis presents work towards the development of an efficient and robust numerical framework for solving reservoir flow problems in a fully-automated manner. In particular, a space-time discontinuous Galerkin (DG) finite element method is used to achieve a high-order discretization on a fully unstructured space-time mesh, instead of a conventional time-marching approach. Anisotropic mesh adaptation is performed to reduce the error of a specified output of interest, by using *a posteriori* error estimates from the dual weighted residual method to drive a metric-based mesh optimization algorithm.

An analysis of the adjoint equations, boundary conditions and solutions of the Buckley-Leverett and two-phase flow equations is presented, with the objective of developing a theoretical understanding of the adjoint behaviors of porous media models. The intuition developed from this analysis is useful for understanding mesh adaptation behaviors in more complex flow problems. This work also presents a new bottom-hole pressure well model for reservoir simulation, which relates the volumetric flow rate of the well to the reservoir pressure through a distributed source term that is independent of the discretization. Unlike Peaceman-type models which require the definition of an equivalent well-bore radius dependent on local grid length scales, this distributed well model is directly applicable to general discretizations on unstructured meshes.

We show that a standard DG diffusive flux discretization of the two-phase flow equations in mass conservation form results in an unstable semi-discrete system in the advection-dominant limit, and hence propose modifications to linearly stabilize the discretization. Further, an artificial viscosity method is presented for the Buckley-Leverett and two-phase flow equations, as a means of mitigating Gibbs oscillations in high-order discretizations and ensuring convergence to physical solutions.

Finally, the proposed adaptive solution framework is demonstrated on compressible two-phase flow problems in homogeneous and heterogeneous reservoirs. Comparisons with conventional time-marching methods show that the adaptive space-time DG method is significantly more efficient at predicting output quantities of interest, in terms of degrees-of-freedom required, execution time and parallel scalability.

Thesis Supervisor: David L. Darmofal
Title: Professor of Aeronautics and Astronautics

Thesis Committee Member: Steven R. Allmaras
Title: Research Engineer of Aeronautics and Astronautics

Thesis Committee Member: Ali H. Dogru
Title: Saudi Aramco Fellow

Thesis Committee Member: Youssef M. Marzouk
Title: Professor of Aeronautics and Astronautics

Acknowledgments

I would like to express my sincere gratitude to all those who have helped, in numerous ways, to make this research study and thesis possible.

First and foremost, I would like to thank my advisor, Prof. David Darmofal, for giving me the opportunity to be a part of his research group, and for his consistent guidance, inspiration and encouragement throughout my graduate study. His insights, critiques and constructive feedback have been vital to the progress of this work. I would also like to thank Dr. Steven Allmaras for his valuable feedback and advice throughout my time at MIT, and for his dedication to the weekly meetings despite being three time zones away. The long chains of emails and discussions we have had about adjoints, boundary conditions and stability have taught me a lot on the subtler aspects of CFD analysis. Dr. Marshall deserves acknowledgement for his significant contributions to the two research codes developed and used by our group, ProjectX and SANS, and for patiently teaching us the ways of writing good code (with unit-tests of course). His vast programming experience and unparalleled availability for debugging, brainstorming and discussing various topics definitely made my graduate life a lot easier. I would also like to thank Nick Burgess and Eric Dow for their valuable input and perspectives from the industry at the weekly meetings, which helped shape the course of this thesis work. I would also like to thank the other members of my thesis committee, Dr. Ali Dogru and Prof. Youssef Marzouk, for their feedback on this thesis work. I'm also grateful to Prof. Abbas Firoozabadi and Prof. Qiqi Wang for reading the initial drafts of this thesis.

This work would have not been possible without the efforts and contributions of the entire ProjectX/SANS teams, both past and present. I would specifically like to thank Arthur, Ben, Carmen, Cory, Hugh, Philip and Shun for their contributions to SANS, and for being a passionate and helpful group of colleagues over the past few years. Masa also deserves recognition for laying down the foundations of the adaptive framework that is used in this work. I would also like to thank all the ACDLers for creating a productive and friendly working environment, including my fellow system

administrators, Chaitanya and Chi, for their support in setting up computers, moving clusters, and helping with all the other sysadmin things that kept the computational resources in the lab running smoothly. Being an ACDL sysadmin was truly a great learning experience outside research. Furthermore, I would also like to thank Beth, Jean, Kate, Meghan and Robin for their help with scheduling meetings, making travel arrangements and all the other important administrative tasks.

Outside the lab, I'm deeply grateful to Charith, Maleen, Sandamali, Nawodya, Prashan, Thulith, Tharanga, and Chamithri for making my life in Boston fun and colorful, with all the game-nights, movies and hikes. Thank you Thanuja, for the Skype calls from the other end of the world, and for being a great friend all these years. Narmada, no words will do justice for the love, support, and wonderful memories you have given me over the past five years. Thank you for making MIT feel home.

Lastly, I would like to thank my parents and sister for their unconditional love and constant support, without which I would never have gotten this far. I'm forever indebted for all the sacrifices you have made to help me pursue my goals. I wish all of you the best of happiness and health.

This research was supported through a Research Agreement with Saudi Aramco, a Founding Member of the MIT Energy Initiative (http://mitei.mit.edu/), with technical monitors Dr. Ali Dogru and Dr. Eric Dow.

Contents

List of Figures

Chapter 1

Introduction

Over the past few decades, numerical simulation has become an indispensable tool for understanding and predicting the behavior of many physical phenomena, ranging from fluid dynamics to electromagnetics. In the context of hydrocarbon reservoirs, numerical simulations are frequently used to investigate flow processes, assess the viability of recovery methods, and predict the overall reservoir performance under different operating conditions. Since the results of these numerical simulations have a significant impact on engineering and management decisions, their accuracy and reliability is of great importance.

1.1 Motivation

A computational fluid dynamics (CFD) method typically utilizes a mesh structure to discretize the domain of the flow, and the numerical flow solution can be interpreted as a distribution of values on this discrete mesh. The resolution of the mesh directly impacts the number of degrees of freedom in the numerical solution, and thereby also the accuracy of the solution. Even with the advances in parallel computing, most large scale reservoir simulators typically solve problems with hundreds of millions of cells, with the most powerful simulators only recently entering the billion-cell regime. For large scale reservoirs which may span tens of kilometers, the size of an average cell in a *mega-cell* model could easily be larger than a city block, inside which all subscale features are averaged out [46]. However, the length scales at which seismic data is provided is typically about an order of magnitude smaller ($\sim$ 25 m) [48], suggesting that existing CFD methods may not be fully utilizing the geological data.

Reservoir performance predictions have improved significantly over the years, primarily driven by the exponential growth of processing power and computer hardware technologies, allowing for ever-increasing mesh resolutions [48, 47, 104]. Finer meshes allow the model to accurately capture localized features such as sharp saturation fronts, gas breakthroughs, and regions of trapped oil, all of which may affect the global performance of the reservoir. However, this approach needs to be done carefully to be cost-effective. For example, increasing the mesh density in regions of smooth flow may not yield significant improvements in accuracy, in comparison to doing so in regions with distinct solution features. Due to the multi-scale nature of the problems, heterogeneity of the geology and the nonlinearity of governing equations, reservoir flows usually contain important local solution features that need to be captured accurately. However, knowing the size, location, and orientation of these features and their impact on the global output of interest is a non-trivial task. As a result, most reservoir simulations performed today require a significant amount of human intervention, particularly during the mesh generation process where the distribution of mesh elements is decided based on "best practices" and expert knowledge of the problem at hand. Furthermore, such "human-in-the-loop" solution processes are known to produce unreliable predictions of engineering outputs, since the engineer's ability to identify all the solution features relevant to the output diminishes as the complexity of the problem increases.

In such cases, where the desired mesh resolution is not known *a priori*, a more attractive alternative is to develop an adaptive method that can autonomously and iteratively modify the mesh, or more generally the discretization, to systematically produce more reliable and accurate output predictions.

1.2 Thesis objective

The objective of this thesis is to develop an efficient and robust solution framework for solving partial differential equations (PDEs) that describe flows in porous media. In particular, this work presents a space-time finite element discretization, coupled with an automated mesh adaptation framework, for accurately predicting output quantities of interest in multi-dimensional, heterogeneous porous media flows.

The proposed solution approach may be viewed as a fusion of three main ideas: space-time methods, solution adaptive methods, and high-order discretizations, with the goal of improving the following aspects of reservoir simulation:

- **Efficiency:** Reducing the amount of computational effort required to produce an output prediction of a given level of accuracy.
- **Autonomy:** Minimizing the amount of user intervention required in the entire PDE solution process, including the mesh generation steps. It is also desirable to isolate the physics of the problem from the numerics, such that users require minimal specialist knowledge of the discretization used.
- **Robustness:** Improving the ability of the solver to produce reliable solutions over a wide range of physical conditions.

1.3 Background

The following sub-sections present a review of the literature pertaining to the work in this thesis.

1.3.1 Space-time adaptive methods

Typically, an unsteady partial differential equation (PDE) is first discretized in space to produce a set of ordinary differential equations that are then discretized in time, following what is often referred to as a method of lines approach. Most reservoir simulations use first or second order accurate temporal discretizations, such as the

Backward Euler method [10, 112, 122]. However, an alternative is to apply the finite element method along the temporal axis as well. The idea of using this “space-time finite element method” dates back to the 1960s, to the work of Oden [106], Argyris and Scharpf [7], and Fried [57].

In a conventional time-marching approach, the ordinary differential equations resulting from the spatial discretization are integrated using the same temporal discretization, producing a *structured* space-time discretization. From a space-time perspective, this is equivalent to using a tensor-product space-time mesh, where each space-time element is a tensor-product of a spatial element and a time-interval. However, as discussed in [76], the potential of the space-time finite element method lies in the use of *unstructured* space-time meshes, where arbitrarily oriented, anisotropic space-time elements can capture solution features more efficiently compared to more constrained tensor-product elements.

Hughes and Hulbert solved the second-order hyperbolic elastodynamic PDE using a space-time method with a continuous Galerkin (CG) method in space and a discontinuous Galerkin (DG) method in time [76, 77]. Their method partitions the space-time domain into decoupled time-slabs, which are solved sequentially by employing the solution at the end of the current time-slab as the initial condition for the next. However, they allow the space-time mesh to be unstructured within each time-slab, as seen in Figure 1-1, making their method attractive for space-time adaptive schemes. More recently, Chen et al. [36] developed a DG method in both space and time to solve a single-phase porous media flow problem using a quadrilateral mesh. In both Hughes and Hulbert and Chen et al, a specific space-time adaptive algorithm is not proposed. In [137], Yano and Darmofal proposed a space-time DG method with fully-unstructured anisotropic mesh adaptation, and demonstrated that it can significantly improve the error-to-degrees-of-freedom efficiency of solving wave-propagation problems for one and two-dimensional spatial domains, compared to tensor-product space-time mesh adaptation. They motivate their method by comparing the number of space-time degrees of freedom (DOF) required to accurately capture an important flow feature of characteristic length $\delta \ll L$, using different types of space-time meshes, where L is the domain length. Assuming the flow feature is transported through space at a constant speed, its motion can be represented by the red lines on the space-time diagrams in Figure 1-2. Their analysis shows that the required space-time DOF scale as $\mathcal{O}(\delta^{-2}), \mathcal{O}(\delta^{-1})$, and $\mathcal{O}(1)$ for the uniformly refined, tensor product, and fully unstructured space-time meshes respectively. The outcome of their simple analysis clearly highlights the potential for large computational savings with space-time adaptive methods, especially for wave propagation problems.

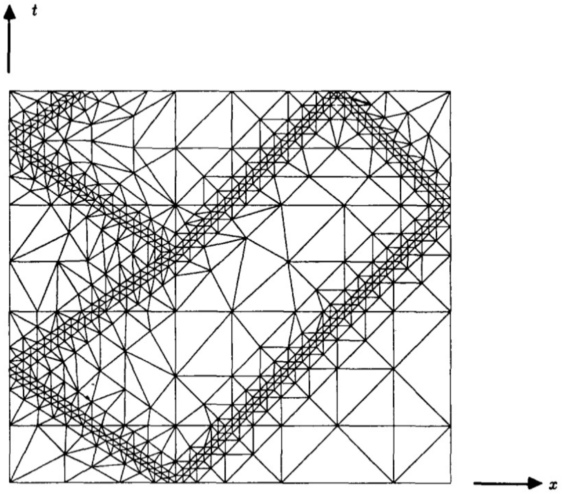
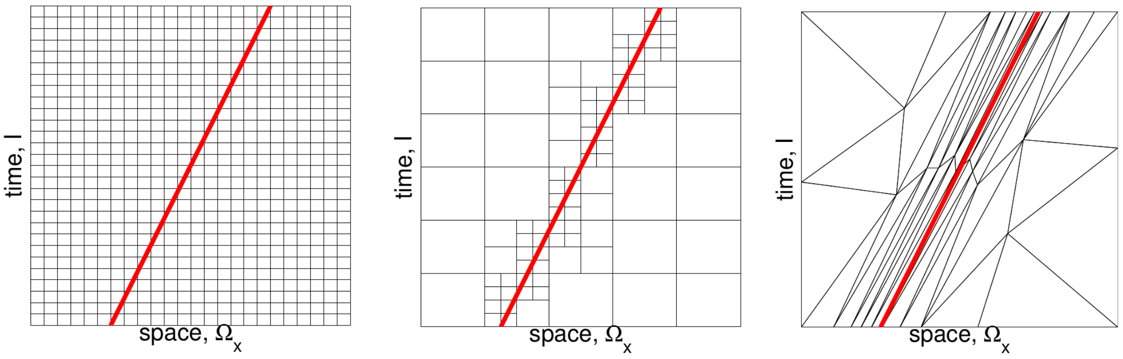


Figure 1-1: Space-time mesh for an elastodynamics problem (Hughes [76])



(a) Uniformly refined

(b) Tensor product

(c) Unstructured

Figure 1-2: Illustration of different space-time meshes

In our previous work, the approach of Yano and Darmofal was extended to porous media flows problems, specifically in the context of reservoir simulations. Fully unstructured space-time mesh adaptation was shown to be significantly more efficient, in terms of output accuracy for a given amount of computational work, for single-phase and two-phase flow problems in 1D spatial domains [78, 79]. The work presented in this thesis is an extension of the same methodology to more realistic flows in 2D spatial domains.

1.3.2 Solution adaptive methods

The objective of a numerical reservoir simulation is to accurately predict outputs of interest, such as the oil recovery factor, oil production rate, or the average pressure in the domain. A solution adaptive numerical method can autonomously arrive at accurate estimates of these outputs of interest, without any prior knowledge of the problem. This minimizes the amount of human intervention required and allows for systematic and reliable output predictions. In this work, this is achieved via *a posteriori* output-based error estimation and mesh adaptation algorithms.

The general outline of the output-based solution adaptation framework can be described using Figure 1-3 as follows. The process begins with a problem statement, which includes the initial mesh, the PDE to be solved, boundary conditions, initial conditions, output function, desired error tolerance and typically a parameter denoting the amount of computational resources available (e.g. $t_{\max}$ = maximum number of CPU hours). The PDE is then solved on this initial mesh and the output error estimates are computed. If the error estimate is larger than the specified tolerance and there is more CPU time available ($t < t_{\max}$), the adaptation algorithm will utilize localized error estimates to generate a new mesh. The process is then repeated with the new adapted mesh until the output error meets the tolerance criterion or the solver runs out of the allocated resources.

A variety of approaches exist for determining where adaptation should occur based upon the solution on the current mesh. For example, the magnitude of solution gradients can be used to identify important features [50, 13, 30, 38, 111]. Other approaches, based on the magnitude of residuals, have been demonstrated for porous media flows by Klieber [85] and Lee [88] using finite element methods, and by Amaziane et al. [4] using the finite volume method. The output-based adaptive method employed in

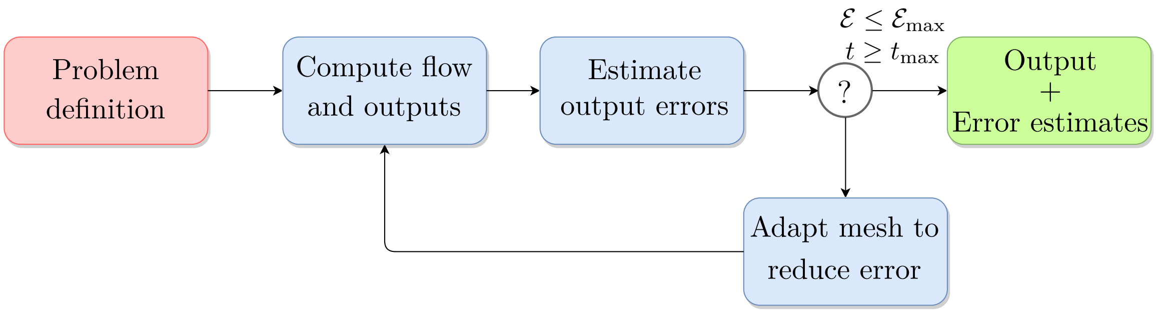


Figure 1-3: General outline of the adaptation framework

this work utilizes the dual-weighted residual (DWR) approach proposed by Becker and Rannacher [24, 25] to obtain both global and local output error estimates, which are then used to drive the mesh adaptation.

This work focuses on h-adaptation, which involves changing the size and shape of elements in the mesh to control the total output error. Isotropic mesh refinement is a widely used mesh adaptation strategy, where selected elements are uniformly refined to decrease the error, as seen in [50, 85, 111, 38, 4, 88] for flows through heterogeneous porous media. However, it is well known that anisotropic mesh adaptation is significantly more efficient for problems involving highly anisotropic features. In this work, we use the Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm proposed by Yano and Darmofal [137, 138] to combine output error estimates with anisotropic adaptation. The MOESS algorithm constructs surrogate error models via element-wise local solves to describe how the output error responds to local changes in the mesh, and then optimizes this error model subject to computational cost constraints to obtain an optimal Riemannian metric tensor field. This metric tensor field, which describes the sizes and orientations of mesh elements, is then passed to a metric-conforming mesh generator to produce a new mesh.

1.3.3 High-order methods

Reservoir simulations are often computed with *low-order* discretizations based on the finite volume method (FVM) [10, 55] and finite difference methods (FDM) [112], where the term “low-order” typically refers to numerical methods that have at most second-order accuracy in space and time [133]. However, in recent years, high-order methods are being applied to porous media flow problems. Finite element methods, such as the DG method, offer a means to obtain high-order accurate solutions by increasing the order of the polynomial basis functions, and have been successfully applied to single-phase [123, 122, 92], two-phase [121, 122, 51, 85, 52, 6, 23], and three-phase [100, 120] flow problems. Additional properties such as local mass conservation on the primal mesh and ease of implementation on unstructured grids make the DG method a competitive alternative to the conventional low-order methods.

The use of a space-time DG discretization in this work also allows for high-order temporal discretizations, without being restricted to the first-order time-marching schemes that are largely used in practice for reservoir simulation. For smooth problems, the higher convergence rates allow high-order methods to achieve a given level of accuracy with fewer degrees of freedom compared to low-order methods [12]. However for problems with low regularity, the efficiency gains of high-order methods may not be realized without also utilizing mesh adaptation.

1.3.4 Adjoint solutions

The adjoint equations to a set of partial differential equations (the primal equations) are useful for computing the sensitivity of an objective function to perturbations in the primal problem. For optimization of PDE-constrained problems, adjoint analysis is an efficient approach to determine the sensitivity of a problem when the number of objective functions and constraints is much smaller than the number of design parameters (controls) [65, 93]. For porous media flows, an important application of adjoint analysis is data assimilation (or history-matching) in which the initial conditions, boundary conditions, and model parameters are adjusted so that the flow solution best matches the available measured data. The optimized primal problem can then be used as the basis of a predictive model for future behavior. Adjoint-based sensitivity analysis methods have been used for performing history matching in single-phase [35, 33, 134, 107], multi-phase [136, 93, 68, 9] and compositional flow problems [54, 86].

Adjoint solutions also play an important role in the analysis and control of numerical errors. The dual-weighted residual (DWR) method developed by Becker and Rannacher is based on the fundamental result that the residual of the approximate primal solution weighted by the adjoint is the error in the objective function [24, 25]. With this insight, Becker and Rannacher developed a grid adaptive method to control a DWR-based estimate of this objective function error. While the DWR method fits most naturally with finite element discretizations, the key ideas have been extended to other discretizations [61, 62, 18]. An extensive literature now exists on a variety of DWR-based adaptive methods applied to a wide range of problems [130, 131, 72, 137, 56, 95, 78, 79].

1.3.5 Well models

Representing the behavior of wells is an important part of the numerical simulation of fluid flows in the subsurface. The large disparity in length scales between a typical well-bore and a reservoir makes it computationally infeasible to explicitly model the near-well pressure behavior by increasing mesh resolution. Therefore, in most practical applications, a mathematical well model is used to capture the interaction between the well-bore and the reservoir, while still allowing the use of grid cells that are a few orders of magnitude larger than the well-bore.

One of the first theoretical studies of wells was done by Peaceman in [113], where a well model is presented for a cell-centered finite difference method on square grids. The analysis provides an interpretation of the well-block pressure and relates it to the flowing bottom-hole pressure of the well, under assumptions of single-phase flow in a homogeneous, isotropic reservoir. Peaceman associates the numerically computed well-block pressure with the steady-state flowing pressure of the actual well at a radial distance r_e away from the well center, which is defined to be the equivalent well radius. Various definitions for the equivalent well radius, including the popular rule of thumb $r_e \approx 0.2h$ where h is the grid spacing, are obtained via numerical experiments and semi-analytic calculations. Peaceman later extended his original well model to allow for non-square Cartesian grids and diagonally anisotropic permeability tensors in

[114], and investigated the effects of off-centered and multiple wells within a well-block in [115]. Abou-Kassem and Aziz [1] also present an analytical approach for computing the equivalent well-bore radius for wells that are located arbitrarily inside the well-block, in a manner that is applicable to both five- and nine-point finite difference schemes in 2D. Peaceman-type well models have also been developed for horizontal wells [11, 110, 58], and inclined wells [89]. Although most well models have been developed for finite difference or finite volume schemes, there also exist a few works which derive well models for the continuous Galerkin finite element (CG), control volume finite element (CVFE) and mixed finite element methods [59, 140]. The recurring theme in all of the well models found in the literature above is the calculation of an equivalent well radius r_e, which is obtained either via tedious mathematical analysis or numerical experiments of a particular numerical discretization. As a result, the derived well models are inherently tied to the specific numerical method and type of mesh that was used to calculate r_e.

Although these methods have been applied to a variety of flow problems with promising results, the literature lacks a generic, rigorously derived well model that can relate the bottom-hole well pressure to the flow rate in a discretization-independent manner. Therefore, most existing works using unstructured meshes, high-order finite element methods or mesh adaptation resort to less attractive approaches for modeling the behavior of wells. One such approach is to impose a Dirichlet boundary condition (BC) for the pressure at the well-bore radius by cutting out the region inside the well-bore from the mesh, as done in [121, 85, 52]. However, this approach is clearly infeasible for large problems since the length scale disparity between a typical well-bore and a reservoir requires an impractical level of mesh resolution in the near-well regions. Furthermore, the presence of “holes” in the mesh increases the complexity of the mesh generation process significantly. One of the contributions of this thesis aims to bridge this gap in the literature, by developing a discretization-independent bottom-hole pressure well model that can be readily used with general discretizations on arbitrarily unstructured meshes, without requiring mesh resolution down to the well-bore.

1.3.6 Shock-capturing methods

High-order linear discretizations produce Gibbs oscillations in regions around discontinuities in the solution (e.g. saturation fronts) and under-resolved features more generally. These unphysical oscillations may propagate and pollute the solution downstream. In the context of multi-phase flows, the oscillations may give rise to unphysical values of saturation and cause the numerical solution to converge to entropy-violating solutions. The goal of shock-capturing methods is to mitigate or eliminate the unphysical oscillations by modifying the discretization through some form of non-linearity. There exists a wide body of literature of such methods, but only a few of the most suitable ones will be reviewed here.

Slope limiters

The goal of using slope limiters (or flux limiters) is to limit solution gradients to physical values, in order to avoid spurious oscillations that may occur in high-order numerical solutions near solution discontinuities. The use of slope limiters make the numerical solutions total variation diminishing (TVD), which implies that no new local extrema are created, the values of local minima do not decrease, and the values of local maxima do not increase. One of the first applications of slope limiters to DG schemes was in a series of papers by Cockburn and Shu, where a Runge-Kutta discontinuous Galerkin method (RKDG) was used with minmod-type slope limiters [41, 40, 39, 43]. In the context of reservoir simulation, slope limiters have been used with DG methods to discretize the saturation equation in two-phase flow problems [34, 102, 75], and the species mass balance equations in compositional flow problems [74, 100]. However, one of the major disadvantages of slope limiting methods is that they do not work well with implicit time-marching schemes, since the clamping of higher-order solution modes near discontinuities and the non-differentiability of most limiters tend to produce ill-conditioned Jacobian matrices, and thereby poor convergence behaviors.

Artificial viscosity

An alternate method is to explicitly add extra dissipation into the problem by introducing diffusion terms to the govering PDE. A simple approach is to increase the amount of physical viscosity in the problem, for example by increasing capillary effects [78], but this has the downside of degrading the accuracy of the solution globally. Hence, a better approach would be to add artificial viscosity in a controlled manner such that the artificial viscosity is zero in smooth regions of the solution and non-zero only in the vicinity of shocks where oscillations occur. Furthermore, the artificial viscosity must vanish as $h \to 0$, in order to ensure the consistency of the numerical discretization.

The amount of artificial viscosity added may be driven by the residuals of the original PDEs or some other predefined *sensor* quantity that is a function of the local solution. The work of Johnson et al [82], Bassi and Rebay [20], and Hartmann and Houston [71], successfully demonstrate the use of residual-based artificial viscosity to control oscillations in finite element solutions of the compressible Euler equations. Similar residual-based methods have been applied to miscible displacement and three-phase flow problems in [128, 127]. The entropy viscosity method introduced in [69] stabilizes the solution by adding an artificial viscosity that is proportional to the entropy residual, and is demonstrated for nonlinear scalar conservation laws and the Euler equations using a continuous Galerkin (CG) method. The entropy viscosity method has also been extended to the DG method in [141], and for miscible displacement problems using the enriched Galerkin method in [88]. The streamline-diffusion shock-capturing (SD-SC) space-time DG method proposed and analyzed by Hiltebrand and Mishra [73] also makes use of a residual-based artificial viscosity operator, and is later modified by Zakerzadeh and May in [139]. The amount of artificial viscosity added can also be controlled by a sensor variable that detects discontinuities and under-resolved regions in the solution. In [118], Persson and Peraire propose an artificial viscosity driven by a discontinuity sensor, which uses the decay rate of higher order solution modes to identify regions with large jumps in the solution. Their method is demonstrated for the 1D Burgers' equation and 2D Euler equations using the DG method. In [101], Moro et al use a dilation-based sensor to compute the artificial viscosity for Navier-Stokes problems, which exploits the presence of strong negative velocity divergences (dilation) near shocks. Barter and Darmofal [17] show that a piecewise-constant artificial viscosity field can introduce spurious oscillations on the gradient of the solution which may corrupt the downstream solution. Hence, they propose a PDE-based artificial viscosity method, where an additional equation is solved in a coupled manner with the original PDE(s) in order to determine the distribution of artificial viscosity over the domain. A reaction-diffusion PDE is used for the auxiliary equation which smoothly diffuses away the artificial viscosity generated by the reaction term. The application of this approach to compressible Navier-Stokes problems demonstrates greater solution accuracy and smoother artificial viscosity distributions compared to piecewise-constant approaches. Unlike slope limiting techniques, artificial viscosity methods obtained by modifying the governing PDE(s) can be discretized using implicit schemes in a straight-forward manner.

Jiang and Shu [81] prove that the standard DG method for scalar conservation laws using Lipschitz continuous monotone fluxes (or E-fluxes) satisfies a cell entropy inequality, but claim that convergence to the unique entropy solution may only be achieved if the flux function is convex. As this is often not true for porous media flow equations (i.e. Buckley-Leverett equation), h-dependent modifications such as streamline diffusion and shock capturing operators need to be added to the DG scheme to show convergence in the case of non-convex flux functions or systems of conservation laws [73, 139].

1.4 Thesis overview

This thesis presents work towards the development of an efficient and robust numerical framework for solving porous media PDEs, based on a space-time DG finite element discretization coupled with output-based mesh adaptation. The primary contributions of this thesis are given below.

- Formulation of a space-time discontinuous Galerkin method for compressible two-phase flow problems in both homogeneous and heterogeneous reservoirs.

- Analysis of the adjoint equations, boundary conditions and analytic adjoint solutions of the Buckley-Leverett and two-phase flow equations, with the goal of developing a theoretical understanding of the adjoint equations and solution behavior.

- Derivation of additional stabilization terms for the standard DG discretization of the two-phase flow equations in mass conservation form, which effectively upwind the underlying saturation equation.

- Development of a distributed bottom-hole pressure well model that is discretization-independent, and therefore applicable to finite element discretizations on unstructured meshes. The proposed well model also includes extensions to anisotropic permeability tensors and multi-phase flows.

- Demonstration of the adaptive space-time DG framework on a slightly compressible two-phase flow problem in a homogeneous reservoir, including performance comparisons with conventional time-marching methods.

- Extension of the PDE-based artificial viscosity method in [17] to the two-phase flow equations, to increase the stability and robustness of the space-time DG method for flows with little or no physical diffusion (capillary effects).

- Demonstration of the adaptive space-time DG method with artificial viscosity on a two-phase problem in a heterogeneous reservoir, with zero capillary effects.

This thesis is organized as follows. Chapter 2 presents the space-time DG method and reviews the DWR method for output error estimation and the MOESS mesh adaptation framework. Chapter 3 presents an analysis of the analytic adjoint equations and solutions of the Buckley-Leverett and two-phase flow equations. Chapter 4 derives the upwinding stabilization terms for the DG discretization of the two-phase flow equations, and presents numerical results for a 1D test problem showing the effect of the stabilization terms on the linear stability of the discretization. Chapter 5 reviews the Peaceman well model and presents a new distributed bottom-hole pressure well model. Numerical results are presented for incompressible single-phase and two-phase flow problems, with comparisons between Peaceman's well model and the distributed well model. Chapter 6 demonstrates the adaptive space-time DG framework on a slightly compressible two-phase flow problem in a homogeneous reservoir, and compares its performance with conventional methods. Chapter 7 presents a PDE-based artificial viscosity method applicable for space-time discretizations of the two-phase flow equations. Chapter 8 demonstrates the adaptive space-time DG method with artificial viscosity on a two-phase flow problem with heterogeneous rock permeabilities, and compares its performance with other approaches. Finally, Chapter 9 summarizes the work presented in this thesis and discusses areas of future work.

Chapter 2

Discretization, Error Estimation, and Output-based Adaptation

This chapter first reviews the space-time discontinuous Galerkin (DG) method for general conservation laws. Then the dual-weighted residual (DWR) method proposed by Becker and Rannacher [24, 25] is presented as a way of estimating the output error. Finally, a summary of the MOESS framework for mesh adaptation presented by Yano and Darmofal [137, 138] is given. Appendix A presents an analysis of the computational costs involved with each of the key steps in the space-time DG mesh adaptation algorithm given below, and shows that the proposed approach scales in a computationally feasible manner to multi-dimensional problems.

2.1 Space-time formulation

Consider a general unsteady conservation law of the form,

$$\frac{\partial}{\partial t}\Big(\mathbf{F}^{\text{temp}}(\mathbf{u})\Big) + \nabla \cdot \Big(\vec{\mathbf{F}}^{\text{adv}}(\mathbf{u}) - \vec{\mathbf{F}}^{\text{diff}}(\mathbf{u}, \nabla\mathbf{u})\Big) + \mathbf{S}(\mathbf{u}, \nabla\mathbf{u}, \vec{x}, t) = \mathbf{0}, \quad \forall \vec{x} \in \Omega_s,\ t \in I \tag{2.1}$$

where $\mathbf{u} \in \mathbb{R}^m$ is the m-variable state vector, $\vec{x}$ represents the spatial coordinates in the d-dimensional spatial domain Ω_s, and t denotes time. $\mathbf{F}^{\text{temp}}(\mathbf{u})$ represents the temporal or unsteady flux, whereas $\vec{\mathbf{F}}^{\text{adv}}(\mathbf{u})$ and $\vec{\mathbf{F}}^{\text{diff}}(\mathbf{u}, \nabla\mathbf{u})$ represent the spatial advective and diffusive fluxes respectively. Any solution-, coordinate- and time-dependent source terms are given by $\mathbf{S}(\mathbf{u}, \nabla\mathbf{u}, \vec{x}, t)$. In a space-time formulation, the d-dimensional unsteady conservation law given above is recast as a $(d+1)$-dimensional conservation law, yielding,

$$\sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} \hat{\mathbf{F}}_j^{\text{adv}}(\mathbf{u}) - \sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} \hat{\mathbf{F}}_j^{\text{diff}}(\mathbf{u}, \hat{\nabla}\mathbf{u}) = \mathbf{S}(\mathbf{u}, \nabla\mathbf{u}, \hat{\vec{x}}), \qquad \forall \hat{\vec{x}} \in \Omega, \tag{2.2}$$

where $\Omega = \Omega_s \cup I \in \mathbb{R}^{d+1}$ is the space-time domain, and $\hat{\vec{x}} = [\vec{x}, t] \in \mathbb{R}^{d+1}$ is the augmented space-time coordinate. The space-time advective flux $\hat{\vec{\mathbf{F}}}^{\text{adv}}(\mathbf{u}) \in \mathbb{R}^{m\times(d+1)}$, and the space-time diffusive flux $\hat{\vec{\mathbf{F}}}^{\text{diff}}(\mathbf{u}, \hat{\nabla}\mathbf{u}) \in \mathbb{R}^{m\times(d+1)}$ can be written in terms of the fluxes in Eq. 2.1 as

$$\hat{\vec{\mathbf{F}}}^{\text{adv}}(\mathbf{u}) = \left[\ \vec{\mathbf{F}}^{\text{adv}}(\mathbf{u}),\ \mathbf{F}^{\text{temp}}(\mathbf{u})\ \right], \tag{2.3}$$

$$\hat{\vec{\mathbf{F}}}^{\text{diff}}(\mathbf{u}, \hat{\nabla}\mathbf{u}) = \left[\ \vec{\mathbf{F}}^{\text{diff}}(\mathbf{u}, \nabla\mathbf{u}),\ \mathbf{0}\ \right]. \tag{2.4}$$

The space-time diffusive flux is also assumed to be a linear function of $\hat{\nabla}\mathbf{u}$, and hence decomposed as,

$$\hat{\vec{\mathbf{F}}}^{\text{diff}}(\mathbf{u}, \hat{\nabla}\mathbf{u}) = \hat{\vec{\mathbf{A}}}(\mathbf{u})\hat{\nabla}\mathbf{u}, \tag{2.5}$$

where $\hat{\vec{\mathbf{A}}}(\mathbf{u})$ is a solution-dependent tensor containing the diffusion coefficients. The boundary conditions are imposed using an operator $\mathcal{B}$ defined as,

$$\mathcal{B}(\mathbf{u}, \hat{\vec{\mathbf{F}}}^{\text{adv}}(\mathbf{u}) \cdot \hat{\vec{n}}, \hat{\vec{\mathbf{F}}}^{\text{diff}}(\mathbf{u}, \hat{\nabla}\mathbf{u}) \cdot \hat{\vec{n}}, \hat{\vec{x}}; BC) = 0, \qquad \forall \hat{\vec{x}} \in \partial\Omega, \tag{2.6}$$

where $\hat{\vec{n}}$ is the space-time unit normal vector pointing out of the domain and BC represents the boundary condition data. The initial condition of the original unsteady conservation law is transformed by the above formulation into a Dirichlet boundary condition at the $t = 0$ boundary of the space-time domain Ω. This "temporal" boundary condition is implemented like any other spatial boundary condition using

$\mathcal{B}$.

Note that in Eqs. 2.2 - 2.6, *hat* accents have been used (i.e. $\hat{\nabla}(\cdot)$) to distinguish $(d+1)$-dimensional *space-time* vectors, fluxes and operators from their d-dimensional spatial counterparts. The rest of this chapter assumes a space-time formulation, hence, the hat accents will be omitted for clarity.

2.2 Space-time DG discretization

The space-time discontinuous Galerkin discretization seeks a solution in a finite dimensional function space $\mathcal{V}_{h,p}$, which is defined as,

$$\mathcal{V}_{h,p} \equiv \left\{ \mathbf{v} \in [L^2(\Omega)]^m : \mathbf{v}|_\kappa \in [\mathcal{P}^p(\kappa)]^m, \forall \kappa \in \mathcal{T}_h \right\}. \tag{2.7}$$

$\mathcal{V}_{h,p}$ represents the piecewise discontinuous solution space of p^{th}-order polynomials on each element of $\mathcal{T}_h$, where $\mathcal{T}_h$ is a triangulation of the space-time domain Ω into non-overlapping elements κ of characteristic size h.

Multiplying Eq. 2.2 by a test function $\mathbf{v}_{h,p} \in \mathcal{V}_{h,p}$ and integrating by parts yields the weak formulation of the governing equation. Solving this weak formulation involves finding a solution $\mathbf{u}_{h,p} \in \mathcal{V}_{h,p}$ that satisfies,

$$\mathcal{R}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}) = 0, \qquad \forall \mathbf{v}_{h,p} \in \mathcal{V}_{h,p}, \tag{2.8}$$

where the semi-linear weighted residual $\mathcal{R}_{h,p} : \mathcal{V}_{h,p} \times \mathcal{V}_{h,p} \to \mathbb{R}$ is composed of three terms,

$$\mathcal{R}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}) = \mathcal{R}^{\text{adv}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}) + \mathcal{R}^{\text{diff}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}) + \mathcal{R}^{\text{source}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}). \tag{2.9}$$

$\mathcal{R}^{\text{adv}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p})$, $\mathcal{R}^{\text{diff}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p})$ and $\mathcal{R}^{\text{source}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p})$ represent the contributions of the advective, diffusive and source terms to the weighted residual, respectively.

2.2.1 Advective flux discretization

The DG discretization of the advective flux term is given by,

$$\begin{aligned}\mathcal{R}^{\mathrm{adv}}_{h,p}(\mathbf{u};\mathbf{v}) = &-\sum_{\kappa\in\mathcal{T}_h}\int_{\kappa}\nabla\mathbf{v}^T\cdot\vec{\mathbf{F}}^{\mathrm{adv}}(\mathbf{u})\;d\Omega \\ &+\sum_{f\in\Gamma_I}\int_f(\mathbf{v}^+-\mathbf{v}^-)^T\mathcal{H}(\mathbf{u}^+,\mathbf{u}^-;\vec{n}^+)\;d\Gamma \\ &+\sum_{f\in\Gamma_B}\int_f{\mathbf{v}^+}^T\mathcal{H}^B(\mathbf{u}^+,\mathbf{u}^B(\mathbf{u}^+;BC);\vec{n}^+)\;d\Gamma,\end{aligned} \tag{2.10}$$

where $(\cdot)^+$ and $(\cdot)^-$ denote the trace values evaluated from opposite sides of a face f and $\vec{n}^+$ is the unit space-time normal vector pointing from the $(+)$ side to the $(-)$ side of a face. Γ_I and Γ_B represent the set of interior and boundary faces in the mesh, respectively. $\mathcal{H}$ and $\mathcal{H}^B$ are the numerical flux functions on the interior and boundary faces, respectively. In this work, $\mathcal{H}$ takes the form,

$$\mathcal{H}(\mathbf{u}^+,\mathbf{u}^-;\vec{n}^+) = \begin{cases}\mathcal{H}_s(\mathbf{u}^+,\mathbf{u}^-;\vec{n}_s^+) + \mathbf{F}^{\mathrm{adv}}_{d+1}(\mathbf{u}^+)\cdot n_t^+, & \text{if } n_t^+ \geq 0,\\ \mathcal{H}_s(\mathbf{u}^+,\mathbf{u}^-;\vec{n}_s^+) + \mathbf{F}^{\mathrm{adv}}_{d+1}(\mathbf{u}^-)\cdot n_t^+, & \text{otherwise,}\end{cases} \tag{2.11}$$

where $\vec{n}_s^+$ and n_t^+ denote the spatial and temporal components of the unit space-time normal vector $\vec{n}^+$. For problems containing spatial advective fluxes, the operator $\mathcal{H}_s$ upwinds the spatial fluxes using a Riemann solver, such as Roe's solver [125] for the Euler or Navier-Stokes equations, or Godunov's exact flux [91] for scalar equations. The advective flux in the temporal direction is evaluated using the solution in the direction of decreasing time (i.e. in the past), in accordance with the laws of causality. At the domain boundaries, the numerical flux $\mathcal{H}^B$ is evaluated using a boundary state $\mathbf{u}^B$, which itself is a function of both the interior state $\mathbf{u}^+$ and the user-specified boundary condition data BC.

2.2.2 Diffusive flux discretization

In most of the previous work where a pressure-saturation formulation of the two-phase flow equations is considered [121, 85, 53, 51], the diffusive fluxes in the pressure equation are discretized using either the Oden-Baumann-Babuska (OBB) method [105], or a generalized form of the non-symmetric interior penalty Galerkin method (NIPG) [124], the symmetric interior penalty Galerkin method (SIPG) [8, 135] and the incomplete interior penalty Galerkin method (IIPG) [44]. In this work, the diffusive flux terms are discretized using the second method proposed by Bassi and Rebay (BR2) [21, 22]. For simplicity of notation, the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators are defined for a scalar s and a vector $\vec{v}$ on an interior face as,

$$\begin{aligned} \{s\} &= \frac{1}{2}(s^+ + s^-), & \{\vec{v}\} &= \frac{1}{2}(\vec{v}^+ + \vec{v}^-), \\ [\![s]\!] &= s^+\vec{n}^+ + s^-\vec{n}^-, & [\![\vec{v}]\!] &= \vec{v}^+ \cdot \vec{n}^+ + \vec{v}^- \cdot \vec{n}^-. \end{aligned} \tag{2.12}$$

The diffusive flux discretization can then be written as follows,

$$\begin{aligned} \mathcal{R}^{\text{diff}}_{h,p}(\mathbf{u};\mathbf{v}) = &\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla\mathbf{v}^T \cdot \left(\vec{\mathbf{A}}(\mathbf{u})\nabla\mathbf{u}\right)\, d\Omega \\ &- \sum_{f\in\Gamma_I}\int_f \left\{\vec{\mathbf{A}}^T(\mathbf{u})\nabla\mathbf{v}\right\}^T \cdot [\![\mathbf{u}]\!]\, d\Gamma \\ &- \sum_{f\in\Gamma_I}\int_f [\![\mathbf{v}]\!]^T \cdot \left\{\vec{\mathbf{A}}(\mathbf{u})\nabla\mathbf{u}\right\}\, d\Gamma \\ &- \sum_{f\in\Gamma_I}\int_f [\![\mathbf{v}]\!]^T \cdot \left\{\vec{\mathbf{A}}(\mathbf{u})\eta_f\vec{\mathbf{r}}_f([\![\mathbf{u}]\!])\right\}\, d\Gamma \\ &- \sum_{f\in\Gamma_B}\int_f \left(\vec{\mathbf{A}}_B^T\nabla\mathbf{v}^+\right)^T \cdot \left(\mathbf{u}^+ - \mathbf{u}^B\right)\vec{n}^+\, d\Gamma \\ &- \sum_{f\in\Gamma_B}\int_f \left(\mathbf{v}^+\vec{n}^+\right)^T \cdot \vec{\mathbf{A}}_B\nabla\mathbf{u}^B\, d\Gamma \\ &- \sum_{f\in\Gamma_B}\int_f \left(\mathbf{v}^+\vec{n}^+\right)^T \cdot \vec{\mathbf{A}}_B\eta_f\vec{\mathbf{r}}_f((\mathbf{u}^+ - \mathbf{u}^B)\vec{n}^+)\, d\Gamma, \end{aligned} \tag{2.13}$$

where the boundary fluxes are set using $\mathbf{u}^B(\mathbf{u}^+; BC)$, $\vec{\mathbf{A}}_B(\mathbf{u}^B; BC)$, and $\nabla\mathbf{u}^B(\nabla\mathbf{u}^+; BC)$. The lifting operator, $\vec{\mathbf{r}}_f : [\mathcal{V}_{h,p}(f)]^{d+1} \to [\mathcal{V}_{h,p}]^{d+1}$, penalizes jumps in the solution across a face, and is defined for an interior face f as,

$$\sum_{\kappa\in\kappa_f}\int_\kappa \vec{\mathbf{s}}^T \cdot \vec{\mathbf{r}}_f(\vec{\mathbf{g}})\, d\Omega = -\int_f \{\vec{\mathbf{s}}\}^T \cdot \vec{\mathbf{g}}\, d\Gamma, \qquad \forall\vec{\mathbf{s}},\vec{\mathbf{g}} \in [\mathcal{V}_{h,p}]^{d+1}, \tag{2.14}$$

where κ_f is the set of elements sharing the face f. For boundary faces, the lifting operator is defined as,

$$\int_{\kappa_B} \vec{\mathbf{s}}^T \cdot \vec{\mathbf{r}}_f(\vec{\mathbf{g}}) \; d\Omega = - \int_f \vec{\mathbf{s}}^{+T} \cdot \vec{\mathbf{g}} \; d\Gamma, \qquad \forall \vec{\mathbf{s}}, \vec{\mathbf{g}} \in [\mathcal{V}_{h,p}]^{d+1}, \tag{2.15}$$

where κ_B is the element containing the boundary face. The stability of the DG discretization requires that the BR2 stabilization parameter, η_f, is greater than or equal to the number of faces in an element [70]. In this work, η_f is set to a slightly conservative value of twice the number of faces in an element, e.g. 6 for triangular meshes and 8 for tetrahedral meshes.

Although standard diffusive flux discretizations such as the BR2 method work well for diffusive problems, they may suffer from stability issues if the diffusive fluxes being discretized are concealing an underlying advection-dominant behavior. In such cases, although counter-intuitive, it is necessary to modify the standard diffusive flux discretization such that the underlying advection-dominant operator is "upwinded". In particular, the two-phase flow equations in mass conservation form contain this complexity, which is discussed and addressed in Chapter 4.

2.2.3 Source discretization

The discretization of the source terms follows the formulation proposed by Bassi *et al.* in [19] where the state gradients are augmented with a global lifting operator as shown below.

$$\mathcal{R}^{\text{source}}_{h,p}(\mathbf{u};\mathbf{v}) = \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \mathbf{v}^T \mathbf{S}(\mathbf{u}, \nabla\mathbf{u} + \vec{\mathbf{r}}_{\text{glob}}(\mathbf{u}), \vec{x}) \; d\Omega, \tag{2.16}$$

where the global lifting operator $\vec{\mathbf{r}}_{\text{glob}} : \mathcal{V}_{h,p} \to [\mathcal{V}_{h,p}]^{d+1}$ is defined as the sum of local lifting operators,

$$\vec{\mathbf{r}}_{\text{glob}}(\mathbf{u}) = \sum_{f \in \Gamma_I} \vec{\mathbf{r}}_f(\llbracket \mathbf{u} \rrbracket) + \sum_{f \in \Gamma_B} \vec{\mathbf{r}}_f \left((\mathbf{u}^+ - \mathbf{u}^B)\vec{n}^+ \right). \tag{2.17}$$

This approach was also shown to be asymptotically dual-consistent by Oliver in [109]. Dual-consistent or asymptotically dual-consistent discretizations have been observed to yield higher convergence rates for an output of interest, compared to dual-inconsistent schemes [108].

2.2.4 Solution method

Expressing the solution $\mathbf{u}_h$ and the test function $\mathbf{v}_h$ in terms of an element-wise discontinuous polynomial basis yields a discrete nonlinear system of equations, which is then solved using Newton's method with a line search algorithm to ensure that residuals decrease. The Jacobian matrices are computed through operator overloaded automatic differentation [60] of the residuals. The ensuing linear systems are solved using the implementation of the restarted generalized minimal residual (GMRES) method [126] given in PETSc [15, 14, 16]. The convergence of the GMRES algorithm is improved by right-preconditioning the linear system using an ILU(k) preconditioner with a minimum discarded fill (MDF) ordering [119]. For parallel solves, the domain is partitioned using ParMETIS [84] and the ILU(k) preconditioner with MDF ordering is applied to each sub-domain, together with a restricted additive Schwarz (RAS) preconditioner [28] with a single layer of overlap for the global system.

All linear systems produced from time-marching discretizations in this work are solved using an ILU(0) preconditioner with the MDF ordering. The space-time DG discretizations in Chapter 6 and 8 are solved with ILU(1) and ILU(2) preconditioners respectively. In all cases, the GMRES algorithm was restarted after 300 iterations.

2.3 Output error estimation

Let the exact value of the output of interest be denoted by,

$$J = \mathcal{J}(\mathbf{u}), \tag{2.18}$$

where $\mathcal{J} : \mathcal{V} \to \mathbb{R}$ is the output functional of interest and $\mathbf{u} \in \mathcal{V}$ is the exact solution to the governing PDE. This is usually expressed as an integral quantity over a surface, such as the mass flow across a boundary, or over a volume, such as the average pressure in the domain. Since the exact solution is not available, an approximation to the exact output can be computed using the discrete DG solution $\mathbf{u}_{h,p} \in \mathcal{V}_{h,p}$ as

$$J_{h,p} = \mathcal{J}_{h,p}(\mathbf{u}_{h,p}), \tag{2.19}$$

where $\mathcal{J}_{h,p} : \mathcal{V}_{h,p} \to \mathbb{R}$ is the discrete output functional. The true error between the exact output and its approximation is given by,

$$\mathcal{E}_{true} = J - J_{h,p} = \mathcal{J}(\mathbf{u}) - \mathcal{J}_{h,p}(\mathbf{u}_{h,p}). \tag{2.20}$$

Since $\mathcal{E}_{true}$ cannot be directly computed in general, the goal of output error estimation is to approximate this true error in the output functional. In this work, the dual-weighted residual (DWR) method proposed by Becker and Rannacher [24, 25] is used.

2.3.1 Dual-weighted residual method

Following the work of Carson *et al.* in [32], a mixed formulation of the weak residual is used for computing the DWR error estimate, which is given by,

$$\begin{aligned}\mathcal{R}_{h,p}(\mathbf{U}_{h,p}; \mathbf{V}_{h,p}) &\equiv \mathcal{R}_{h,p}((\mathbf{u}_{h,p}, \vec{\mathbf{r}}_{h,p}); (\mathbf{v}_{h,p}, \vec{\mathbf{s}}_{h,p}))\\ &= \mathcal{R}^{\text{adv}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p}) + \mathcal{R}^{\text{diff}}_{h,p}(\mathbf{u}_{h,p}, \vec{\mathbf{r}}_{h,p}; \mathbf{v}_{h,p}) + \mathcal{R}^{\text{source}}_{h,p}(\mathbf{u}_{h,p}; \mathbf{v}_{h,p})\\ &\quad + \mathcal{R}^{\text{lift}}_{h,p}(\mathbf{u}_{h,p}, \vec{\mathbf{r}}_{h,p}; \vec{\mathbf{s}}_{h,p}),\end{aligned} \tag{2.21}$$

where the definition of the lifting operators on the interior and boundary faces is appended to the weak form residual given in Eq. 2.9 via $\mathcal{R}^{\text{lift}}_{h,p}$, which is given by,

$$\mathcal{R}^{\text{lift}}_{h,p}(\mathbf{u}, \vec{\mathbf{r}}; \vec{\mathbf{s}}) = -\sum_{f \in \Gamma_I} \left(\sum_{\kappa \in \kappa_f} \int_\kappa \vec{\mathbf{s}}_f^T \cdot \vec{\mathbf{r}}_f \; d\Omega + \int_f \{\vec{\mathbf{s}}_f\}^T \cdot [\![\mathbf{u}]\!] \; d\Gamma \right)$$

$$- \sum_{f \in \Gamma_B} \left(\int_{\kappa_B} \vec{\mathbf{s}}_f^T \cdot \vec{\mathbf{r}}_f \; d\Omega + \int_f \{\vec{\mathbf{s}}_f\}^T \cdot (\mathbf{u}^+ - \mathbf{u}^B)\vec{n}^+ \; d\Gamma \right), \qquad (2.22)$$

where $\vec{\mathbf{r}}$ and $\vec{\mathbf{s}} \in \mathcal{S}_{h,p}$, where $\mathcal{S}_{h,p}$ is the collective space of the lifting operators on all faces, defined as $\mathcal{S}_{h,p} \equiv \prod_{f \in \Gamma_I \cup \Gamma_B} [\mathcal{V}_{h,p}]^{d+1}$. $\vec{\mathbf{r}}_f$ and $\vec{\mathbf{s}}_f$ are the elements corresponding to face f in $\vec{\mathbf{r}}$ and $\vec{\mathbf{s}}$ respectively. The $\mathcal{R}_{h,p}^{\text{diff}}$ operator in Eq. 2.21 is the BR2 diffusive flux operator described in Section 2.2.2, but with modified arguments to accept the lifting operators as an input. For notational simplicity, the arguments of $\mathcal{R}_{h,p}$ are specified as the tuples $\mathbf{U}_{h,p} = (\mathbf{u}_{h,p}, \vec{\mathbf{r}}_{h,p}), \mathbf{V}_{h,p} = (\mathbf{v}_{h,p}, \vec{\mathbf{s}}_{h,p}) \in \mathcal{V}_{h,p} \times \mathcal{S}_{h,p}$. The corresponding infinite dimensional tuples are given by $\mathbf{U} = (\mathbf{u}, \vec{\mathbf{r}}), \mathbf{V} = (\mathbf{v}, \vec{\mathbf{s}}) \in \mathcal{V} \times \mathcal{S}$.

Then, the DWR method represents the true output error as follows,

$$\mathcal{E}_{true} = \mathcal{J}(\mathbf{u}) - \mathcal{J}_{h,p}(\mathbf{u}_{h,p}) = -\mathcal{R}_{h,p}(\mathbf{U}_{h,p}; \boldsymbol{\Psi}), \qquad (2.23)$$

where $\boldsymbol{\Psi} = (\boldsymbol{\psi}, \vec{\boldsymbol{\sigma}}) \in \mathcal{W} \equiv (\mathcal{V} + \mathcal{V}_{h,p}) \times (\mathcal{S} + \mathcal{S}_{h,p})$ is the true adjoint tuple that consists of the primary adjoint ($\boldsymbol{\psi}$), and the lifting operator adjoint ($\vec{\boldsymbol{\sigma}}$) respectively. It is worth noting that the true lifting operator $\vec{\mathbf{r}}$ is identically zero, and that similarly, the true lifting operator adjoint $\vec{\boldsymbol{\sigma}}$ is zero for volume integral outputs over Ω [32]. The adjoint solutions satisfy the following dual problem,

$$\overline{\mathcal{R}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}](\mathbf{W}; \boldsymbol{\Psi}) = \overline{\mathcal{J}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}](\mathbf{w}), \qquad \forall \mathbf{W} \equiv (\mathbf{w}, \vec{\boldsymbol{\xi}}) \in \mathcal{W}, \qquad (2.24)$$

where $\overline{\mathcal{R}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}] : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ and $\overline{\mathcal{J}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}] : (\mathcal{V} + \mathcal{V}_{h,p}) \to \mathbb{R}$ are the mean-value linearizations defined as,

$$\overline{\mathcal{R}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}](\mathbf{W}; \mathbf{V}) \equiv \int_0^1 \mathcal{R}'_{h,p}\left[(1-\theta)\mathbf{U} + \theta \mathbf{U}_{h,p}\right](\mathbf{W}; \mathbf{V}) \; d\theta, \qquad (2.25)$$

$$\overline{\mathcal{J}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}](\mathbf{w}) \equiv \int_0^1 \mathcal{J}'_{h,p}\left[(1-\theta)\mathbf{U} + \theta \mathbf{U}_{h,p}\right](\mathbf{w}) \; d\theta. \qquad (2.26)$$

$\mathcal{R}'_{h,p}[\mathbf{z}](\cdot,\cdot)$ and $\mathcal{J}'_{h,p}[\mathbf{z}](\cdot)$ denote the Fréchet derivatives of $\mathcal{R}_{h,p}(\cdot,\cdot)$ and $\mathcal{J}_{h,p}(\cdot)$ with respect to the first argument, evaluated about $\mathbf{z}$.

The true output error may also be expressed using the definition of the mean-value linearized residual as

$$\mathcal{E}_{true} = -\overline{\mathcal{R}}'_{h,p}[\mathbf{U}, \mathbf{U}_{h,p}](\mathbf{U} - \mathbf{U}_{h,p}; \boldsymbol{\Psi} - \boldsymbol{\Psi}_{h,p}), \tag{2.27}$$

which shows that the true output error is a function of the error in the primal solution, $\mathbf{U} - \mathbf{U}_{h,p}$, as well as the error in the adjoint solution, $\boldsymbol{\Psi} - \boldsymbol{\Psi}_{h,p}$.

The true adjoint $\boldsymbol{\Psi} = (\boldsymbol{\psi}, \vec{\boldsymbol{\sigma}})$ is not computable in general since it lives in an infinite dimensional space $\mathcal{W}$, and its computation requires the true primal solution $\mathbf{U}$. Hence, the true adjoint solution is approximated by a finite dimensional adjoint $\boldsymbol{\Psi}_{h,\hat{p}} = (\boldsymbol{\psi}_{h,\hat{p}}, \vec{\boldsymbol{\sigma}}_{h,\hat{p}}) \in \mathcal{V}_{h,\hat{p}} \times \mathcal{S}_{h,\hat{p}}$ (for $\hat{p} > p$) which is obtained by solving the following dual problem linearized about $\mathbf{U}_{h,p}$,

$$\mathcal{R}'_{h,\hat{p}}[\mathbf{U}_{h,p}](\mathbf{V}_{h,\hat{p}}; \boldsymbol{\Psi}_{h,\hat{p}}) = \mathcal{J}'_{h,\hat{p}}[\mathbf{U}_{h,p}](\mathbf{v}_{h,\hat{p}}), \quad \forall \mathbf{V}_{h,\hat{p}} \in \mathcal{V}_{h,\hat{p}} \times \mathcal{S}_{h,\hat{p}}. \tag{2.28}$$

It is worth noting that the lifting operators $\vec{\mathbf{r}}_{h,p}$ and the lifting operator adjoints $\vec{\boldsymbol{\sigma}}_{h,\hat{p}}$ are computed via post-processing steps (i.e. static condensation), and hence are not solved as a part of the global system of unknowns.

The DWR error estimate of the output is obtained by substituting this approximate adjoint into Eq. 2.23,

$$\mathcal{E}_{true} \approx -\mathcal{R}_{h,p}(\mathbf{U}_{h,p}; \boldsymbol{\Psi}_{h,\hat{p}}). \tag{2.29}$$

The approximate adjoint $\boldsymbol{\Psi}_{h,\hat{p}}$ needs to exist in a space that is richer than that of the approximate primal solution $\mathbf{U}_{h,p}$ (i.e. $\mathcal{V}_{h,\hat{p}} \supset \mathcal{V}_{h,p}$), else the DWR estimate yields zero due to Galerkin orthogonality. In this work, the polynomial order of the adjoint approximation is chosen to be one order higher than that of the primal solution, i.e. $\hat{p} = p + 1$.

Error localization

A global estimate of the output error is not sufficient for mesh adaptation since it needs to identify regions in the domain with large and small contributions to the error. Therefore, a localized error estimate η_κ, associated with element κ, is obtained by an element-wise restriction of the adjoint weight as follows,

$$\eta_\kappa \equiv \left| \mathcal{R}_{h,p}(\mathbf{U}_{h,p}; \boldsymbol{\Psi}_{h,\hat{p}}|_\kappa) \right| . \tag{2.30}$$

A bound of the error estimate can be obtained by summing the local error estimates over all elements,

$$\mathcal{E} \equiv \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa . \tag{2.31}$$

The localized error estimate η_κ is a conservative estimate due to the presence of the absolute value operator in Eq. 2.30. If this absolute value operator is omitted, the sum of local error estimates over all elements will recover the global error estimate given in Eq. 2.29.

2.4 Mesh adaptation

The goal of mesh adaptation is to use localized output error estimate information to produce a new mesh that achieves a lower output error. This is frequently done by performing isotropic mesh refinement on the current mesh, where elements in selected regions are either uniformly refined or de-refined according to their contribution to the total output error [85, 4, 38]. But reservoir flows often contain features such as saturation fronts and jumps in pressure gradients caused by discontinuities in geological properties, which can be captured more efficiently with anisotropic elements than with isotropic elements. Therefore the mesh adaptation algorithm needs to be able to represent and manipulate the anisotropy of elements in the mesh, in addition to a measure of their size h.

2.4.1 Continuous mesh framework

The anisotropic information of a space-time element κ can be represented using a metric tensor $\mathcal{M}_\kappa$, which is a $(d+1)\times(d+1)$ symmetric positive definite (SPD) matrix [131]. This metric tensor can be interpreted as a straight-forward extension of the scalar valued element size h, which not only contains a measure of the element's size, but also its orientation. By collecting the elemental metric tensors, $\{\mathcal{M}_\kappa\}_{\kappa\in\mathcal{T}_h}$, a continuous Riemannian metric field $\{\mathcal{M}(\vec{x})\}_{\vec{x}\in\Omega}$ can be constructed. A metric-conforming triangulation is a triangulation where all the edges are close to unit length as measured under the Riemannian metric field $\{\mathcal{M}(\vec{x})\}_{\vec{x}\in\Omega}$. The length of a segment $\overrightarrow{ab}$ from point $a\in\Omega$ to point $b\in\Omega$ under the metric is given by,

$$l_{\mathcal{M}}(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^T\mathcal{M}(a+\overrightarrow{ab}s)\overrightarrow{ab}}\; ds. \tag{2.32}$$

Note that this length measure reduces to the standard Euclidean distance if the metric $\mathcal{M}$ is the identity tensor.

An example of a metric-mesh pair is given in Figure 2-1, where the metric tensor field is illustrated by ellipses. Relying on the geometric duality between the discrete mesh and the Riemannian metric field, Yano shows that the polynomial approximation errors and the output errors incurred on a metric-conforming discrete mesh are a function of the Riemannian metric field from which the discrete mesh was generated [137, 138]. This key result allows the development of a mesh adaptation algorithm that attempts to decrease the output error by optimizing a continuous metric tensor field, instead of a discrete mesh.

2.4.2 Mesh Optimization via Error Sampling and Synthesis

This sub-section contains a short review of the MOESS algorithm developed by Yano and Darmofal [137, 138], which is used for mesh adaptation in this work.

The objective of the mesh adaptation algorithm is to manipulate the current triangulation $\mathcal{T}_h$ to reduce the errors in output predictions. A formal statement of

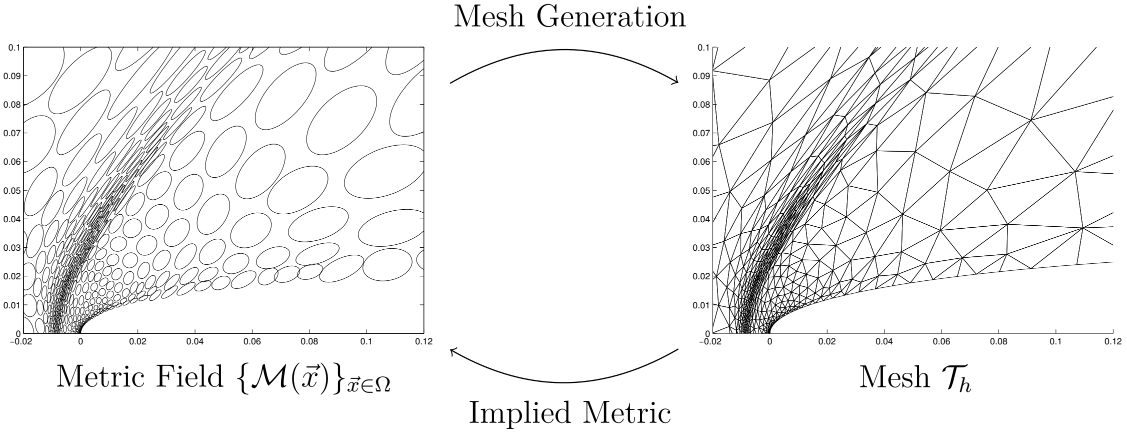


Figure 2-1: Mesh metric-field duality (Modisette [99])

this problem involves finding the optimal triangulation $\mathcal{T}_h^*$ given by,

$$\mathcal{T}_h^* = \arg\inf_{\mathcal{T}_h} \mathcal{E}(\mathcal{T}_h) \qquad \text{s.t.} \qquad \mathcal{C}(\mathcal{T}_h) \leq C_{\max}, \tag{2.33}$$

where $\mathcal{E}$ is the error functional that represents the output error incurred by solving on $\mathcal{T}_h$, and $\mathcal{C}$ is the cost functional that represents the cost of solving on $\mathcal{T}_h$. In this work, the cost is taken to be the number of degrees of freedom in $\mathcal{T}_h$, and therefore $C_{\max}$ is the maximum number of DOF allowed in the solution, which is usually set by the amount of available computational resources. Since the triangulation $\mathcal{T}_h$ is defined by node coordinates and node connectivity, the optimization problem presented above is a discrete-continuous optimization problem, and is generally intractable.

However, an approximate solution to the problem can be found by considering the continuous relaxation of the discrete problem, as proposed by Loseille and Alauzet [96], where a continuous Riemannian metric field, $\boldsymbol{\mathcal{M}} \equiv \{\mathcal{M}(\vec{x})\}_{\vec{x}\in\Omega}$ is optimized instead of the discrete mesh. This relaxed optimization problem involves finding an optimal metric field, $\boldsymbol{\mathcal{M}}^*$, where,

$$\boldsymbol{\mathcal{M}}^* = \arg\inf_{\boldsymbol{\mathcal{M}}} \mathcal{E}(\boldsymbol{\mathcal{M}}) \qquad \text{s.t.} \qquad \mathcal{C}(\boldsymbol{\mathcal{M}}) \leq C_{\max}. \tag{2.34}$$

In order to apply a DOF constraint, the cost functional $\mathcal{C}(\boldsymbol{\mathcal{M}})$ is defined as,

$$\mathcal{C}(\boldsymbol{\mathcal{M}}) = \int_{\Omega} c_p \sqrt{\det(\mathcal{M}(\vec{x}))} d\vec{x}, \tag{2.35}$$

where c_p is the number of degrees of freedom in the reference element, normalized by its size. Furthermore, it is assumed that the total error is the sum of the elementwise local error contributions η_κ, and that each local contribution η_κ is also a function of the elemental metric tensor $\mathcal{M}_\kappa$. The error functional $\mathcal{E}$ can then be approximated as,

$$\mathcal{E}(\boldsymbol{\mathcal{M}}) \approx \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa(\mathcal{M}_\kappa). \tag{2.36}$$

To complete the problem statement, a definition of the local error function $\eta_\kappa(\mathcal{M}_\kappa)$ is needed, but since their form is not known *a priori*, surrogate models of these functions are constructed via a sampling procedure on each element.

Local error sampling

The objective of the local error surrogate model is to capture how changes to an element's configuration affects its output error contribution. This surrogate model is constructed by solving local problems with different local configurations of a given element, and then recalculating the local error estimate associated with each configuration. In particular, for each space-time element $\kappa_0 \in \Omega$, let there be a set of new local configurations, $\{\kappa_i\}_{i=1}^{n_{\text{config}}}$, each of which is obtained by splitting one or multiple edges of κ_0. Figure 2-2 shows an example of the split configurations used for a triangle element, and the implied metric tensors $\mathcal{M}_{\kappa_i}$ associated with each configuration.

For each split configuration κ_i, an element-wise local problem is solved to find the local solution $\mathbf{u}_{h,p}^{\kappa_i} \in \mathcal{V}_{h,p}(\kappa_i)$ that satisfies,

$$\mathcal{R}_{h,p}^{\kappa_i}(\mathbf{u}_{h,p}^{\kappa_i}; \mathbf{v}_{h,p}^{\kappa_i}) = 0, \qquad \forall \mathbf{v}_{h,p}^{\kappa_i} \in \mathcal{V}_{h,p}(\kappa_i), \tag{2.37}$$

where the local semi-linear form $\mathcal{R}_{h,p}^{\kappa_i}(\cdot, \cdot)$ imposes boundary fluxes on κ_i by assuming

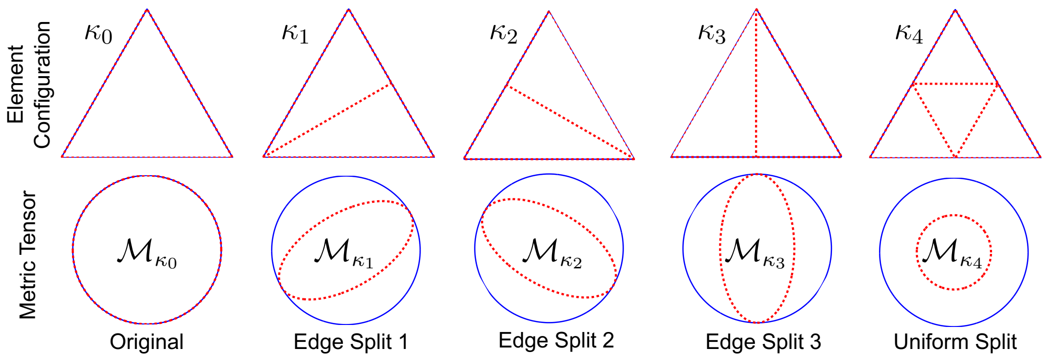


Figure 2-2: Example split configurations with associated metric tensors (Yano [137])

that the solution on neighboring elements does not change. Next, a localized DWR error estimate associated with the configuration κ_i is computed as,

$$\eta_{\kappa_i} \equiv \left| \mathcal{R}_{h,p}^{\kappa_i}(\mathbf{U}_{h,p}^{\kappa_i}; \mathbf{\Psi}_{h,\hat{p}}|_{\kappa_0}) \right| \tag{2.38}$$

where $\mathbf{U}_{h,p}^{\kappa_i} = (\mathbf{u}_{h,p}^{\kappa_i}, \mathbf{r}_{h,p}^{\kappa_i})$ and $\hat{p} = p + 1$ as before. Finally, the local metric $\mathcal{M}_{\kappa_i}$ associated with configuration κ_i is obtained by computing an affine-invariant average of the implied metric tensors of each sub-element in κ_i [116]. The set of metric-error pairs, $\{\mathcal{M}_{\kappa_i}, \eta_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$, computed using this local sampling procedure can then be synthesized to form a continuous local error model.

Local error model synthesis

The continuous metric-error function $\eta_\kappa(\cdot) : \text{Sym}_{d+1}^{+} \to \mathbb{R}^{+}$ aims to capture how the local error is affected by local changes to the metric field. For this purpose, a symmetric "step" tensor, $S_\kappa(\mathcal{M}_\kappa)$, is defined to characterize the change in the metric tensor from configuration κ_0 to some new configuration κ. This measurement is based on Pennec's affine-invariant framework [116] as follows,

$$S_\kappa(\mathcal{M}_\kappa) \equiv \log\left(\mathcal{M}_{\kappa_0}^{-\frac{1}{2}} \mathcal{M}_\kappa \mathcal{M}_{\kappa_0}^{-\frac{1}{2}} \right), \tag{2.39}$$

where $\log(\cdot)$ is the matrix logarithm. Note that the above function maps the metric of the original configuration $\mathcal{M}_{\kappa_0}$ to the zero tensor (i.e. $S_{\kappa_0} = 0$). Similarly, a measure of the change in error between configurations is also defined as,

$$f_{\kappa_i} \equiv \log\left(\frac{\eta_{\kappa_i}}{\eta_{\kappa_0}}\right). \tag{2.40}$$

The information from the pairs $\{S_{\kappa_i}, f_{\kappa_i}\}_{i=1}^{n_{\text{config}}}$ is then synthesized to construct a continuous function $f_\kappa(\cdot) : \text{Sym}_{d+1} \rightarrow \mathbb{R}$, which is assumed to be of the linear form,

$$f_\kappa(S_\kappa) = \text{tr}(R_\kappa S_\kappa), \tag{2.41}$$

where $R_\kappa \in \text{Sym}_{d+1}$ is a "rate" tensor that is synthesized from the known data by performing a least-squares regression as follows,

$$R_\kappa = \arg \min_{Q \in \text{Sym}_{d+1}} \sum_{i=1}^{n_{\text{config}}} \left(f_{\kappa_i} - \text{tr}(Q S_{\kappa_i}(\mathcal{M}_{\kappa_i}))\right)^2. \tag{2.42}$$

The rate tensor R_κ can be thought of as a generalization of the convergence rate for isotropic scaling to anisotropic changes. Finally, the local error model can be written in terms of a step tensor S_κ as,

$$\eta_\kappa(\mathcal{M}_\kappa(S_\kappa)) = \eta_{\kappa_0} \exp(\text{tr}(R_\kappa S_\kappa)). \tag{2.43}$$

Metric optimization

The final step of the adaptation process is to optimize the Riemannian metric field $\{\mathcal{M}(x)\}_{x\in\Omega}$ using the error and cost models constructed above. Since the metric field is described by the vertex values $\{\mathcal{M}_v\}_{v\in\mathcal{V}}$, the objective is to find the vertex step matrices, $\{S_v\}_{v\in\mathcal{V}}$, that minimize the error functional. $S_v \in \text{Sym}_{d+1}$ represents the change in the metric that is required at vertex v.

Upon formulating the error objective function and the cost constraints in terms of the design variables $\{S_v\}_{v\in\mathcal{V}}$, the constrained optimization problem given in Eq. 2.34 is solved using a gradient-based optimization algorithm as shown by Kudo [87]. This work utilizes the globally-convergent method-of-moving-asymptotes (MMA) algorithm [129] implemented in NLopt [83] to solve the optimization problem.

The resulting optimal vertex-based metric field is then passed to a metric-conforming mesh generator to generate a new mesh. All adapted meshes (2D and 3D) used in this thesis were generated using the `FeFlo.a` mesher developed by Loseille and Löhner [97, 98].

Chapter 3

Adjoint analysis of the Buckley-Leverett and two-phase flow equations

Adjoint solutions play an important role in the dual-weighted residual (DWR) method used in this work for estimating output errors. Since the error estimates are obtained by replacing the continuous adjoint solution with a discrete adjoint solution, as shown in Section 2.3.1, it is imperative that the numerical method used to compute the discrete adjoint solution employs an *adjoint-consistent* formulation, which ensures that the discrete adjoint problem is a consistent discretization of the continuous adjoint problem.

Although the discrete adjoint method (via the linearization of the discrete residual operator) works for general problems, it does not necessarily provide a clear insight into the nature of the adjoint solution. An analytic approach can be used to provide a theoretical understanding of the adjoint PDE, boundary conditions, and solution behavior, which can then also be used to verify discrete adjoint solutions on simplified problems. The work in this chapter is motivated by the desire for a theoretical understanding of the adjoint equations for representative models of porous media flows. Specifically, the focus is on the Buckley-Leverett equation and a two-equation two-phase flow model. The results of this chapter were published in [80].

3.1 Scalar conservation laws with shocks

This section presents a derivation of 1D scalar conservation law adjoint equations for different output functional types. These general results are later specialized to the case of the Buckley-Leverett equation and compared against numerical results. Consider the 1D scalar conservation law given in Eq. (3.1), with the initial and boundary conditions given by Eq. (3.2) - (3.3).

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{3.1}$$

$$u(x,0) = u_0(x), \qquad x \in [0,L] \tag{3.2}$$

$$u(0,t) = u_L(t), \qquad t \in [0,T] \tag{3.3}$$

Without loss of generality, characteristics are assumed to enter the domain from the left boundary at all times ($\frac{\partial f}{\partial u} > 0$ at $x = 0$). If the solution $u(x,t)$ contains shocks, then the Rankine-Hugoniot jump condition,

$$\dot{x}_s \llbracket u \rrbracket - \llbracket f \rrbracket = 0, \tag{3.4}$$

must be satisfied, where $x_s(t)$ and $\dot{x}_s$ represent the spatial location and speed of the shock respectively. The jump operator in 1D, defined as $\llbracket \cdot \rrbracket = (\cdot)^+ - (\cdot)^-$, represents the jump in a certain quantity between the left $(+)$ and right $(-)$ sides of the shock. The primal problem described by Eq. (3.1) - (3.4) is represented in the following space-time form,

$$\nabla \cdot \vec{F} = 0, \qquad \vec{x} \in \Omega, \tag{3.5}$$

where $\vec{x} = (x,t)$ is the augmented space-time coordinate, $\Omega = [0,L] \times [0,T]$ is the space-time domain, and $\vec{F}$ represents the space-time fluxes,

$$\vec{F} = (F_x(u), F_t(u)) = (f(u), u)\,. \tag{3.6}$$

The Rankine-Hugoniot jump condition given in Eq. (3.4) transforms to the equivalent jump condition in space-time, given by,

$$\left[\!\left[-\vec{F} \right]\!\right] = 0, \qquad \vec{x} \in \Gamma_s, \tag{3.7}$$

where Γ_s is the curve that tracks the path of the shock, and the jump operator definition has been extended to multiple dimensions for scalar and vector quantities as follows,

$$[\![s]\!] = s^+ \vec{n}^+ + s^- \vec{n}^- = (s^+ - s^-)\vec{n}^+, \tag{3.8}$$

$$[\![\vec{v}]\!] = \vec{v}^+ \cdot \vec{n}^+ + \vec{v}^- \cdot \vec{n}^- = (\vec{v}^+ - \vec{v}^-) \cdot \vec{n}^+, \tag{3.9}$$

where $\vec{n}^+$ is the space-time unit normal vector pointing from the left to the right of interface Γ_s, and $\vec{n}^- = -\vec{n}^+$. The components of the space-time unit normal vector $\vec{n}^+ = \left(n_x^+, n_t^+\right)$ depend on the shock speed as follows,

$$n_x^+ = \frac{1}{\sqrt{\dot{x}_s^2 + 1}}, \qquad n_t^+ = \frac{-\dot{x}_s}{\sqrt{\dot{x}_s^2 + 1}}. \tag{3.10}$$

A schematic of the space-time domain and the shock path is given in Figure 3-1.

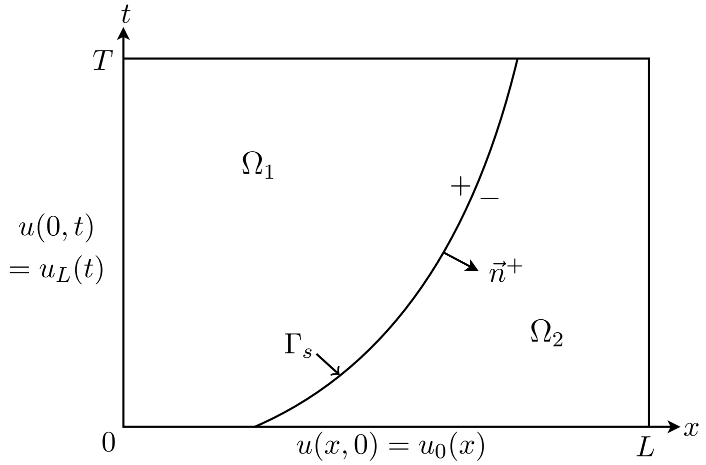


Figure 3-1: Schematic of space-time domain Ω

Let Ω_1 and Ω_2 be partitions of the space-time domain to the left and right of the shock respectively, separated by the interface Γ_s as shown in Figure 3-1. The boundaries of Ω_1 and Ω_2, including Γ_s, are denoted by Γ_1 and Γ_2 respectively. Next, consider the weak form of the primal equations in $\Omega_1 \cup \Omega_2$ and the Rankine-Hugoniot relation across Γ_s,

$$R(u, x_s; w, w_s) = \int_{\Omega_1 \cup \Omega_2} w \nabla \cdot \vec{F} d\Omega - \int_{\Gamma_s} w_s \left[\!\left[\vec{F} \right]\!\right] d\Gamma, \tag{3.11}$$

where w and w_s are admissible test functions. The linearized form of Eq. (3.11) is obtained by considering infinitesimal perturbations of the solution, denoted by $\delta u(x,t)$, and the shock location, denoted by $\delta x_s(t)$. Perturbing the shock location by $\delta x_s(t)$ results in a horizontal perturbation of the shock interface Γ_s by a vector $\vec{\delta}_s = (\delta x_s(t), 0)$. The resulting perturbed weak form is given by Eq. (3.12),

$$\begin{aligned} R + \delta R = & \int_{\Omega_1 \cup \Omega_2} w \nabla \cdot (\vec{F} + \delta \vec{F}) d\Omega \\ & + \int_{\Gamma_s} w^+ (\nabla \cdot \vec{F}^+) \vec{\delta}_s \cdot \vec{n}^+ d\Gamma - \int_{\Gamma_s} w^- (\nabla \cdot \vec{F}^-) \vec{\delta}_s \cdot \vec{n}^+ d\Gamma \\ & - \int_{\Gamma_s} w_s \left[\!\left[(\vec{F} + \delta \vec{F}) \right]\!\right] d\Gamma - \int_{\Gamma_s} w_s \left(\vec{F}^+ - \vec{F}^- \right) \cdot \delta \vec{n}^+ \, d\Gamma. \end{aligned} \tag{3.12}$$

Using the definition of R in Eq. (3.11) to cancel out terms, and rewriting in terms of the jump operator yields,

$$\begin{aligned} \delta R = & \int_{\Omega_1 \cup \Omega_2} w \nabla \cdot (\delta \vec{F}) d\Omega + \int_{\Gamma_s} \left[\!\left[w (\nabla \cdot \vec{F}) \right]\!\right] \cdot \vec{\delta}_s \, d\Gamma \\ & - \int_{\Gamma_s} w_s \left[\!\left[\delta \vec{F} \right]\!\right] d\Gamma - \int_{\Gamma_s} w_s \left(\vec{F}^+ - \vec{F}^- \right) \cdot \delta \vec{n}^+ \, d\Gamma. \end{aligned} \tag{3.13}$$

Note that the second integral in Eq. (3.13) vanishes since $\nabla \cdot \vec{F} = 0$. Invoking the chain rule to represent the flux perturbations $\delta \vec{F}$ in terms of δu and δx_s gives,

$$\begin{aligned} \delta R = & \int_{\Omega_1 \cup \Omega_2} w \nabla \cdot \left(\vec{A} \delta u \right) d\Omega - \int_{\Gamma_s} w_s \left[\!\left[\left(\vec{A} \delta u \right) + \frac{\partial \vec{F}}{\partial x} \delta x_s \right]\!\right] d\Gamma \\ & - \int_{\Gamma_s} w_s \left(\vec{F}^+ - \vec{F}^- \right) \cdot \delta \vec{n}^+ \, d\Gamma, \end{aligned} \tag{3.14}$$

where $\vec{A} = \frac{\partial \vec{F}}{\partial u}$. Performing integration by parts on the volume integral yields,

$$\begin{aligned}\delta R = &- \int_{\Omega_1 \cup \Omega_2} \nabla w \cdot \left(\vec{A}\delta u\right) d\Omega + \int_{\Gamma_1 \cup \Gamma_2} w \left(\vec{A}\delta u\right) \cdot \vec{n}\ d\Gamma \\ &- \int_{\Gamma_s} w_s \left[\!\!\left[\left(\vec{A}\delta u\right) + \frac{\partial \vec{F}}{\partial x}\delta x_s \right]\!\!\right] d\Gamma - \int_{\Gamma_s} w_s \left(\vec{F}^+ - \vec{F}^-\right) \cdot \delta \vec{n}^+\ d\Gamma.\end{aligned} \tag{3.15}$$

The boundary integral in Eq. (3.15) is separated into an integral over the shock interface Γ_s and an integral over the domain boundary, $\Gamma_B = (\Gamma_1 \cup \Gamma_2)\backslash\Gamma_s$, as shown below,

$$\begin{aligned}\delta R = &- \int_{\Omega_1 \cup \Omega_2} \nabla w \cdot \left(\vec{A}\delta u\right) d\Omega + \int_{\Gamma_B} w \left(\vec{A}\delta u\right) \cdot \vec{n}\ d\Gamma \\ &+ \int_{\Gamma_s} \left(\left[\!\!\left[w \left(\vec{A}\delta u\right) \right]\!\!\right] - w_s \left[\!\!\left[\left(\vec{A}\delta u\right) \right]\!\!\right] \right)\ d\Gamma \\ &- \int_{\Gamma_s} w_s \left(\left[\!\!\left[\frac{\partial \vec{F}}{\partial x}\delta x_s \right]\!\!\right] + \left(\vec{F}^+ - \vec{F}^-\right) \cdot \delta \vec{n}^+ \right) d\Gamma.\end{aligned} \tag{3.16}$$

The expression inside the brackets in the last integral of Eq. (3.16) is simplified using the approach outlined in Appendix B, resulting in the following equation for δR,

$$\begin{aligned}\delta R(\delta u, \delta x_s; w, w_s) = &- \int_{\Omega_1 \cup \Omega_2} \nabla w \cdot \left(\vec{A}\delta u\right) d\Omega + \int_{\Gamma_B} w \left(\vec{A}\delta u\right) \cdot \vec{n}\ d\Gamma \\ &+ \int_{\Gamma_s} \left(\left[\!\!\left[w \left(\vec{A}\delta u\right) \right]\!\!\right] - w_s \left[\!\!\left[\left(\vec{A}\delta u\right) \right]\!\!\right] \right)\ d\Gamma \\ &+ \int_{\Gamma_s} w_s \frac{d}{dt} \left((F_t^+ - F_t^-)\delta x_s \right) n_x^+ d\Gamma.\end{aligned} \tag{3.17}$$

Given a generic output functional $J(u)$ and its linearization $\delta J(\delta u, \delta x_s)$, the adjoint solutions ψ and ψ_s satisfy the following equation for all $\delta u, \delta x_s$ [64],

$$\delta R(\delta u, \delta x_s; \psi, \psi_s) = \delta J(\delta u, \delta x_s). \tag{3.18}$$

The relationship of these adjoint solutions to the calculation of output sensitivities, as required for inverse analysis and design optimization, is described in Appendix C. The following sub-sections formulate the adjoint equation and boundary conditions for two different output functionals.

3.1.1 Output: spatial integral at $t = T$

This section assumes that the output functional of interest is the spatial integral of some solution dependent quantity $g(u)$ at $t = T$,

$$J_T = \int_0^L g(u(x,T))dx. \tag{3.19}$$

Splitting the output into integrals to the left and right of the shock and linearizing gives,

$$\delta J_T(\delta u, \delta x_s) = \int_0^{x_s(T)} \frac{\partial g}{\partial u}\delta u(x,T)\ dx + \int_{x_s(T)}^{L} \frac{\partial g}{\partial u}\delta u(x,T)\ dx + [\![g]\!]_{t=T}\,\delta x_s(T), \tag{3.20}$$

where $[\![g]\!]_{t=T}$ represents the jump in the value of g across the shock at the final time T. The adjoint definition (Eq. (3.18)) with this output yields:

$$\begin{aligned} -\int_{\Omega_1\cup\Omega_2} \nabla\psi \cdot \left(\vec{A}\delta u\right) d\Omega + \int_{\Gamma_B} \psi \left(\vec{A}\delta u\right)\cdot \vec{n}\ d\Gamma \\ + \int_{\Gamma_s} \left(\left[\!\!\left[\psi\left(\vec{A}\delta u\right)\right]\!\!\right] - \psi_s \left[\!\!\left[\left(\vec{A}\delta u\right)\right]\!\!\right]\right)\ d\Gamma \\ + \int_{\Gamma_s} \psi_s \frac{d}{dt}\left((F_t^+ - F_t^-)\delta x_s\right) n_x^+ d\Gamma = \int_0^{x_s(T)} \frac{\partial g}{\partial u}\delta u(x,T)\ dx \\ + \int_{x_s(T)}^{L} \frac{\partial g}{\partial u}\delta u(x,T)\ dx \\ + [\![g]\!]_{t=T}\,\delta x_s(T). \end{aligned} \tag{3.21}$$

The adjoint PDE is obtained by equating volume integrals on both sides of Eq. (3.21) and noting that the resulting equation is valid for any perturbation δu.

$$-\int_{\Omega_1\cup\Omega_2} \nabla\psi \cdot \left(\vec{A}\delta u\right) d\Omega = 0, \tag{3.22}$$

$$\boxed{\nabla\psi \cdot \frac{\partial \vec{F}}{\partial u} = 0.} \tag{3.23}$$

The adjoint boundary conditions are obtained by collecting the domain boundary integrals in Eq. (3.21). All domain boundary integrals at $t = 0$ and $x = 0$ vanish since the primal initial condition and left boundary condition requires $\delta u(x,0)$ and $\delta(0,t)$ to be zero, respectively. As a result, there are no adjoint boundary conditions at the bottom ($t=0$) and left ($x=0$) boundaries. The absence of a primal boundary condition at the right ($x=L$) boundary implies that $\delta u(L,t) \neq 0$, hence requiring the following adjoint boundary condition in order for the boundary integrals at $x=L$ to satisfy Eq. (3.21),

$$\boxed{\psi(L,t) = 0.} \tag{3.24}$$

The boundary integrals at $t=T$ give,

$$\begin{aligned} &\int_0^{x_s(T)} \psi(x,T)\frac{\partial F_t}{\partial u}\delta u(x,T)\ dx + \int_{x_s(T)}^{L} \psi(x,T)\frac{\partial F_t}{\partial u}\delta u(x,T)\ dx \\ &= \int_0^{x_s(T)} \frac{\partial g}{\partial u}\delta u(x,T)\ dx + \int_{x_s(T)}^{L} \frac{\partial g}{\partial u}\delta u(x,T)\ dx. \end{aligned} \tag{3.25}$$

Requiring Eq. (3.25) to hold for any perturbation $\delta u(x,T)$ yields the following adjoint boundary condition at $t=T$,

$$\boxed{\psi(x,T)\frac{\partial F_t}{\partial u}(x,T) = \frac{\partial g}{\partial u}(x,T), \qquad \forall x \neq x_s(T).} \tag{3.26}$$

The behavior of the adjoint variables at the shock is found by analyzing the shock interface integrals in Eq. (3.21). Collecting all shock interface integrals that depend on δx_s gives,

$$\int_{\Gamma_s} \psi_s \frac{d}{dt}\Big((F_t^+ - F_t^-)\delta x_s\Big)\, n_x^+ d\Gamma = [\![g]\!]_{t=T}\, \delta x_s(T). \tag{3.27}$$

Performing integration by parts in time using $d\Gamma = dt/n_x^+$, and noting that $\delta x_s(0)=0$ due to the primal initial condition yields,

$$\psi_s(T)\,[\![F_t]\!]_{t=T}\,\delta x_s(T) - \int_0^T \frac{d\psi_s}{dt}(F_t^+ - F_t^-)\delta x_s dt = [\![g]\!]_{t=T}\,\delta x_s(T), \tag{3.28}$$

where $[\![F_t]\!]_{t=T}$ is the jump in F_t across the shock at time T. Requiring Eq. (3.28) to hold for any $\delta x_s(t)$ gives the following conditions for $\psi_s(t)$,

$$\psi_s(T)\,[\![F_t]\!]_{t=T} = [\![g]\!]_{t=T}\,, \tag{3.29}$$

$$\frac{d\psi_s}{dt} = 0. \tag{3.30}$$

Solving the ordinary differential equation (ODE) above shows that ψ_s is a constant with the following value,

$$\boxed{\psi_s = \frac{[\![g]\!]_{t=T}}{[\![F_t]\!]_{t=T}}.} \tag{3.31}$$

Lastly, the third integral on the left-hand side of Eq. (3.21) gives the following condition across the shock,

$$\left[\!\!\left[\psi\left(\vec{A}\delta u\right)\right]\!\!\right] = \psi_s\left[\!\!\left[\left(\vec{A}\delta u\right)\right]\!\!\right]. \tag{3.32}$$

Expanding all components of Eq. (3.32) using the definition of $\vec{A}$ yields,

$$\boxed{\left(\psi^+ - \psi_s\right)\left(\frac{\partial F_x}{\partial u}^+ - \dot{x}_s\frac{\partial F_t}{\partial u}^+\right)\delta u^+ - \left(\psi^- - \psi_s\right)\left(\frac{\partial F_x}{\partial u}^- - \dot{x}_s\frac{\partial F_t}{\partial u}^-\right)\delta u^- = 0.} \tag{3.33}$$

Conditions on ψ^+, ψ^- and ψ_s are obtained by analyzing the nature of the terms in Eq. (3.33). If $\left(\frac{\partial F_x}{\partial u}^+ - \dot{x}_s\frac{\partial F_t}{\partial u}^+\right)$ is non-zero, then $\psi^+ = \psi_s$ satisfies Eq. (3.33) for any variation δu^+. By the same argument, $\psi^- = \psi_s$, if $\left(\frac{\partial F_x}{\partial u}^- - \dot{x}_s\frac{\partial F_t}{\partial u}^-\right)$ is non-zero. This is the case for the Burgers' equation, where the adjoint is continuous across the shock (i.e. $\psi^+ = \psi_s = \psi^-$) [3].

However, if $\left(\frac{\partial F_x}{\partial u}^+ - \dot{x}_s\frac{\partial F_t}{\partial u}^+\right)$ or $\left(\frac{\partial F_x}{\partial u}^- - \dot{x}_s\frac{\partial F_t}{\partial u}^-\right)$ is identically zero for a particular set of primal fluxes, then the equality of ψ^+ and ψ_s, or ψ^- and ψ_s respectively, cannot be inferred from Eq. (3.33) alone. In particular, the Buckley-Leverett equation contains this complexity, and Section 3.2 gives a more detailed analysis of Eq. (3.33) in this context.

3.1.2 Output: volume integral over space-time domain

This section assumes that the output functional of interest is the integral of some solution dependent quantity $g(u)$ over the entire space-time domain,

$$J = \int_\Omega g(u) d\Omega. \tag{3.34}$$

The linearized output variation is given by:

$$\delta J(\delta u, \delta x_s) = \int_{\Omega_1 \cup \Omega_2} \frac{\partial g}{\partial u} \delta u \; d\Omega + \int_{\Gamma_s} [\![g]\!] \cdot \vec{\delta}_s d\Gamma. \tag{3.35}$$

Using the same approach as in Section 3.1.1, the adjoint definition given by Eq. (3.18) yields,

$$\begin{aligned} -\int_{\Omega_1 \cup \Omega_2} \nabla \psi \cdot \left(\vec{A} \delta u \right) d\Omega + \int_{\Gamma_B} \psi \left(\vec{A} \delta u \right) \cdot \vec{n} \; d\Gamma \\ + \int_{\Gamma_s} \left(\left[\!\!\left[\psi \left(\vec{A} \delta u \right) \right]\!\!\right] - \psi_s \left[\!\!\left[\left(\vec{A} \delta u \right) \right]\!\!\right] \right) d\Gamma \\ + \int_{\Gamma_s} \psi_s \frac{d}{dt} \left((F_t^+ - F_t^-) \delta x_s \right) n_x^+ d\Gamma = \int_{\Omega_1 \cup \Omega_2} \frac{\partial g}{\partial u} \delta u \; d\Omega + \int_{\Gamma_s} [\![g]\!] \cdot \vec{\delta}_s d\Gamma. \end{aligned} \tag{3.36}$$

From this, the adjoint PDE is inferred to be,

$$\boxed{\nabla \psi \cdot \frac{\partial \vec{F}}{\partial u} = -\frac{\partial g}{\partial u}.} \tag{3.37}$$

The adjoint boundary conditions are determined as before, by following the discussion in Section 3.1.1. However, the change in output functional gives a different adjoint BC at $t = T$,

$$\boxed{\psi(x, T) = 0, \qquad \forall x \neq x_s(T).} \tag{3.38}$$

Manipulating the integrals along the shock in Eq. (3.36) gives the following ODE for $\psi_s(t)$,

$$\boxed{\frac{d\psi_s}{dt} = -\frac{g^+ - g^-}{F_t^+ - F_t^-},} \tag{3.39}$$

subject to the condition,

$$\boxed{\psi_s(T) = 0.} \tag{3.40}$$

3.2 Buckley-Leverett equation

This section applies the results of Section 3.1 to the case of the Buckley-Leverett problem,

$$\frac{\partial}{\partial t}\left(\phi S_w\right) + \frac{\partial}{\partial x}\left(u_T f_w(S_w)\right) = 0, \tag{3.41}$$

$$S_w(x,0) = 0.1, \quad x \in [0,L] \tag{3.42}$$

$$S_w(0,t) = 1, \qquad t \in [0,T] \tag{3.43}$$

where the wetting phase saturation S_w is the dependent variable, porosity $\phi = 0.3$, and total velocity $u_T = 0.3$ ft/day. S_w is a non-dimensional quantity that takes physical values in the range $[0,1]$. The fractional flow function $f_w(S_w)$ [10] is a nonlinear, non-convex function defined as,

$$f_w(S_w) = \frac{S_w^2}{S_w^2 + \frac{\mu_w}{\mu_n}(1-S_w)^2}. \tag{3.44}$$

In this work, the wetting-phase to non-wetting phase viscosity ratio $\frac{\mu_w}{\mu_n}$ is assumed to be equal to 0.5, and the relative permeabilities are modeled as quadratic functions. The domain length L is equal to 50 ft, and the final time T is 25 days. The space-time fluxes for this PDE are,

$$\vec{F} = \left[F_x(S_w), F_t(S_w)\right] = \left[u_T f_w(S_w),\ \phi S_w\right]. \tag{3.45}$$

The solution to this particular problem is a combined rarefaction-shock wave that originates at $x = 0$. The downstream state of the shock is given by the initial saturation value in the domain,

$$S_w(x_s^-, t) = 0.1. \tag{3.46}$$

The upstream state of the shock is obtained by solving the following nonlinear problem, which equates the characteristic speed on the upstream state of the shock to the shock speed given by the Rankine-Hugoniot jump condition,

$$\frac{u_T}{\phi}\frac{df_w(S_w^+)}{dS_w} = \frac{u_T}{\phi}\left(\frac{f_w(S_w^+) - f_w(S_w^-)}{S_w^+ - S_w^-}\right)$$

$$\frac{df_w(S_w^+)}{dS_w} = \frac{f_w(S_w^+) - f_w(0.1)}{S_w^+ - 0.1}, \tag{3.47}$$

$$S_w(x_s^+, t) = \frac{\sqrt{249} - 3}{24} \approx 0.53249. \tag{3.48}$$

The corresponding shock speed is given by,

$$\dot{x}_s(t) = \frac{u_T}{\phi}\frac{\partial f_w}{\partial S_w}^+ = 1.61324 \text{ ft/day}. \tag{3.49}$$

Figure 3-2 contains a plot of the primal space-time solution obtained using a second-order discontinuous Galerkin (DG) finite element method, on a structured triangular space-time mesh with $750,000$ degrees of freedom (DOF). Figure 3-3 shows the familiar Buckley-Leverett saturation front propagating to the right at a constant speed, obtained from constant-time slices of the space-time solution in Figure 3-2. The numerical solutions from the space-time DG method (solid lines) agree well with the analytical solution (dashed lines). This figure clearly shows the compound wave behavior of the Buckley-Leverett solution, where a rarefaction wave is observed upstream (to the left) of the propagating shock.

Figure 3-4 depicts characteristic paths of the Buckley-Leverett problem defined above. The characteristic paths downstream of the shock either end at the shock (blue region), or leave the domain through the top ($t = T$) and right ($x = L$) boundaries (grey and red regions respectively). Upstream of the shock, all characteristics leave the top boundary. The equality of the limiting upstream characteristic speed and the shock speed (Eq. (3.49)) causes the limiting upstream characteristic to be parallel to the shock front.

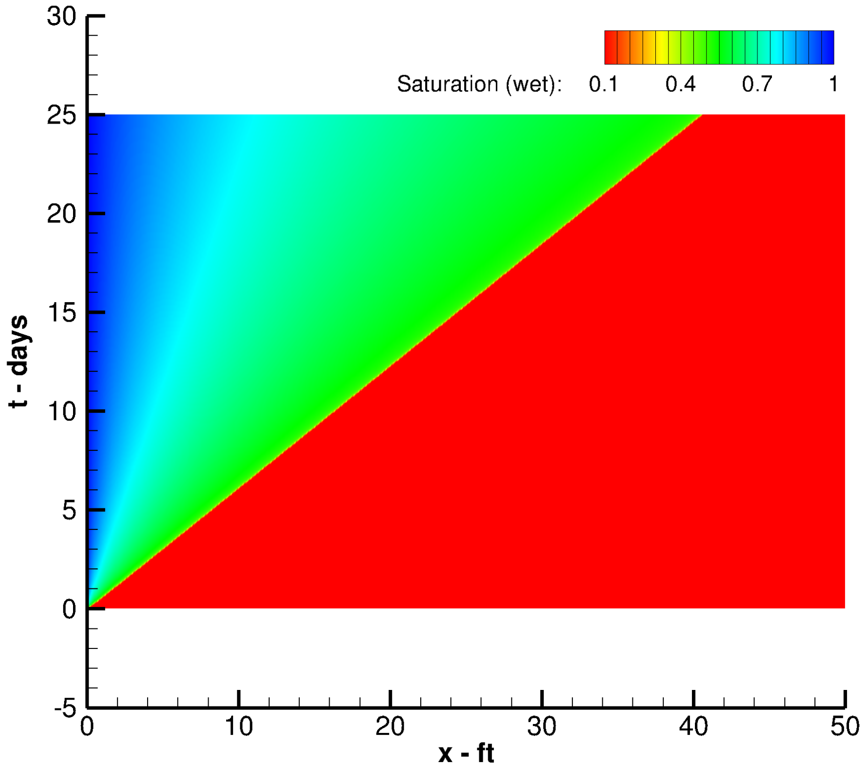


Figure 3-2: Primal solution of Buckley-Leverett problem using a second-order space-time DG method with 750,000 DOF.

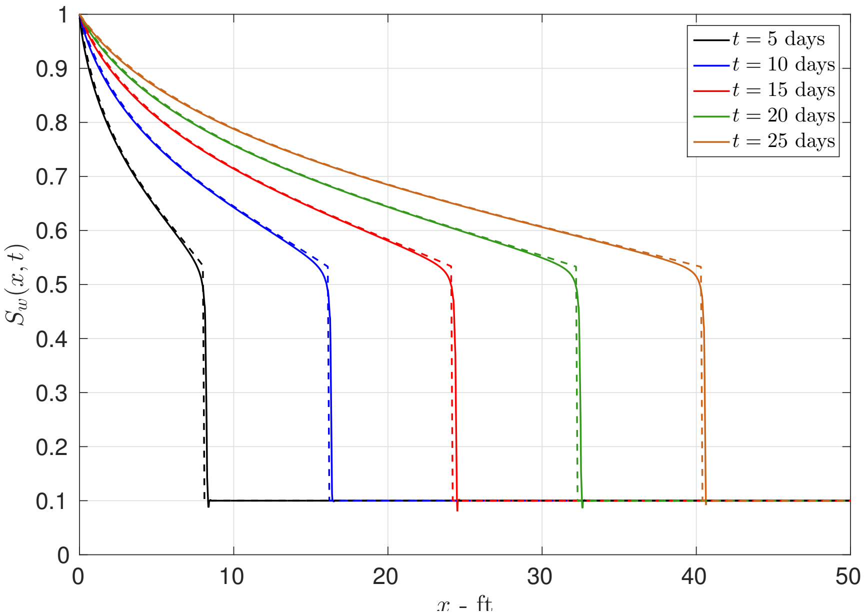


Figure 3-3: Comparison of space-time DG (solid lines) and exact (dashed lines) primal solutions at different times.

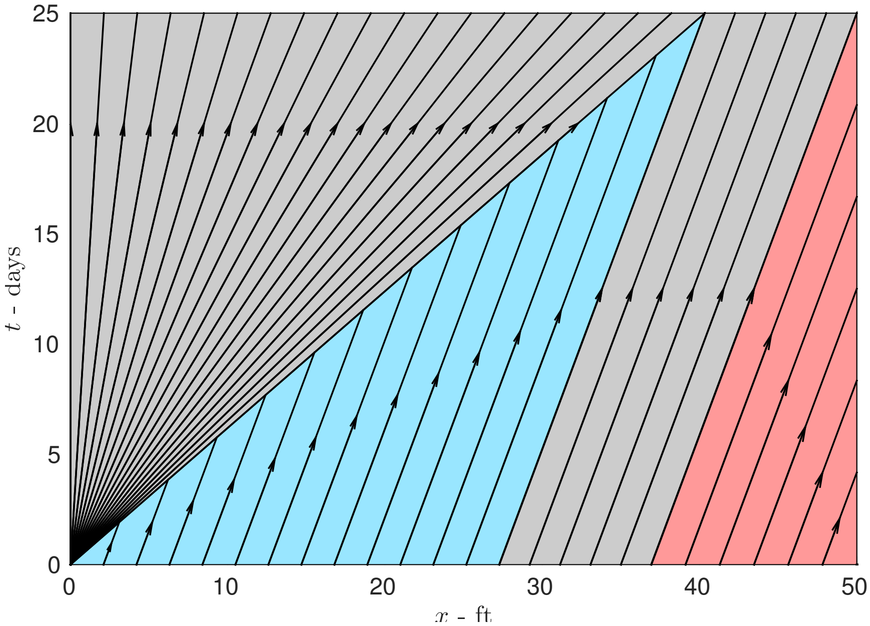


Figure 3-4: Primal characteristics of the Buckley-Leverett problem entering the shock (blue region) or exiting the top (grey region) and right (red region) boundaries.

3.2.1 Output: spatial integral at $t = T$

This section presents the adjoint problem and its solution for the Buckley-Leverett problem defined above, for the output functional given in Eq. (3.50),

$$J_T = \int_0^L g(S_w(x,T))dx = \int_0^L S_w^2(x,T)dx. \tag{3.50}$$

Using Eq. (3.23), the adjoint equation for this problem is

$$\phi\frac{\partial\psi}{\partial t} + \left(u_T\frac{\partial f_w}{\partial S_w}\right)\frac{\partial\psi}{\partial x} = 0. \tag{3.51}$$

Following the discussion on boundary conditions in Section 3.1.1, no adjoint boundary conditions are required at the left or bottom boundaries. The boundary conditions at the right and top boundaries follow from Eq. (3.24) and (3.26) respectively,

$$\psi(L,t) = 0, \qquad \forall t \in [0,T] \tag{3.52}$$

$$\psi(x,T) = \frac{2S_w(x,T)}{\phi}, \qquad \forall x \neq x_s(T). \tag{3.53}$$

The value of ψ_s is computed from Eq. (3.31),

$$\psi_s = \frac{[\![g]\!]_{t=T}}{[\![\phi S_w]\!]_{t=T}} = \frac{1}{\phi}\left(S_w(x_s^+, T) + S_w(x_s^-, T)\right). \tag{3.54}$$

The analytical values of S_w on either side of the shock, given previously, reduce Eq. (3.54) to

$$\psi_s = \psi(x_s^-, t) = \frac{1}{36}(5\sqrt{249} - 3) \approx 2.10830. \tag{3.55}$$

Finally, Eq. (3.33) gives

$$\left(\psi^+ - \psi_s\right)\left(u_T \frac{\partial f_w}{\partial S_w}^+ - \phi \dot{x}_s\right)\delta u^+ - \left(\psi^- - \psi_s\right)\left(u_T \frac{\partial f_w}{\partial S_w}^- - \phi \dot{x}_s\right)\delta u^- = 0. \tag{3.56}$$

Since the upstream characteristic speed converges to the shock speed (Eq. (3.49)), the upstream flux term in Eq. (3.56) vanishes, yielding,

$$\left(\psi(x_s^-, t) - \psi_s\right)\left(u_T \frac{\partial f_w}{\partial S_w}^- - \phi \dot{x}_s\right)\delta u^- = 0. \tag{3.57}$$

Recognizing that the characteristic speed to the right of the shock does not generally match the shock speed, and requiring Eq. (3.57) to hold for any δu^- gives the following condition on the adjoint:

$$\psi(x_s^-, t) = \psi_s(t). \tag{3.58}$$

Eq. (3.56) cannot give a relationship between $\psi(x_s^+, t)$ and $\psi_s(t)$ because the first term vanishes, which means that these two quantities differ by an arbitrary amount. However, by using the method of characteristics, the value of $\psi(x_s^+, t)$ is obtained by tracing the characteristic path to the top ($t = T$) boundary, where the value of ψ is given by Eq. (3.53). Note that this result differs from the usual result obtained for PDEs with convex fluxes, such as the Burgers' equation, where characteristics flow into the shock from both sides causing the adjoint variable to be continuous across the shock (i.e. $\psi(x_s^+, t) = \psi(x_s^-, t) = \psi_s(t)$) [66, 63]. However, the rarefaction-shock

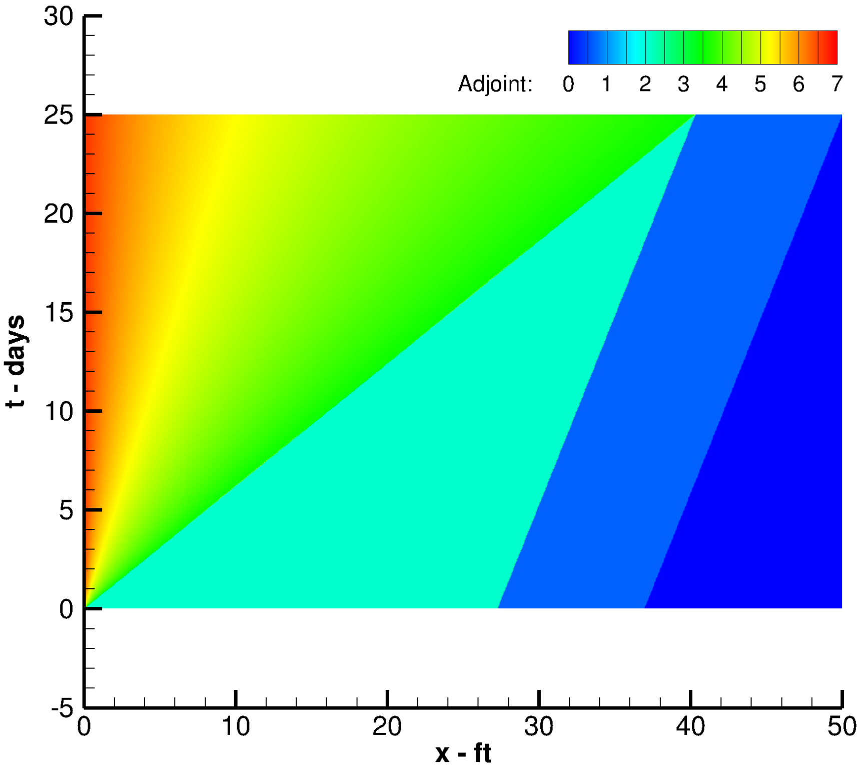


Figure 3-5: Exact adjoint solution for output J_T.

behavior of the Buckley-Leverett equation causes this property to no longer hold, allowing a finite jump between $\psi(x_s^+, t)$ and $\psi_s(t)$.

Figure 3-5 shows a contour plot of the analytical adjoint solution in the space-time domain, computed by analyzing the characteristics of the adjoint equation in Eq. (3.51) (as outlined in Appendix C of [80]). Figure 3-6 contains the same plot with a numerical adjoint solution, obtained by a second-order space-time DG finite element method, on a structured triangular mesh with $750,000$ degrees of freedom.

The adjoint solution has a constant value of ψ_s along all characteristics emanating from the shock. Furthermore, the absence of a source term in the adjoint PDE (Eq. (3.51)) means that $\psi(x,t)$ is also constant along each characteristic that emanates from the top and right boundaries. Figure 3-7 compares the DG adjoint solutions (solid lines) at different times, with the corresponding exact solutions (dashed lines). The numerical results agree well with the analytical solutions, with the largest errors occuring around discontinuities as a result of numerical diffusion.

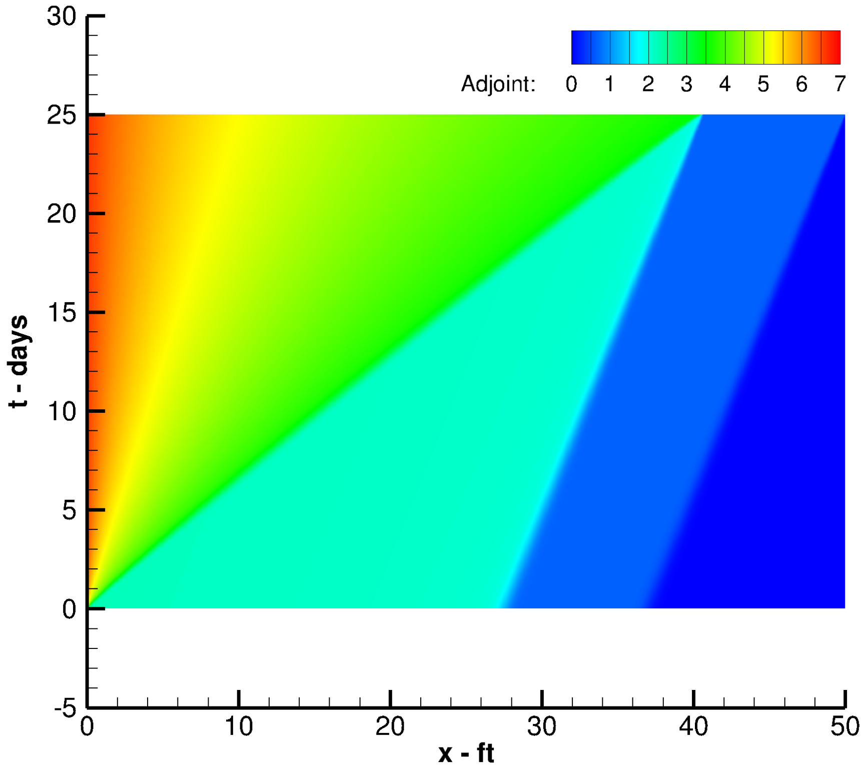


Figure 3-6: Numerical adjoint solution for output J_T from a second-order space-time DG method with 750,000 DOF.

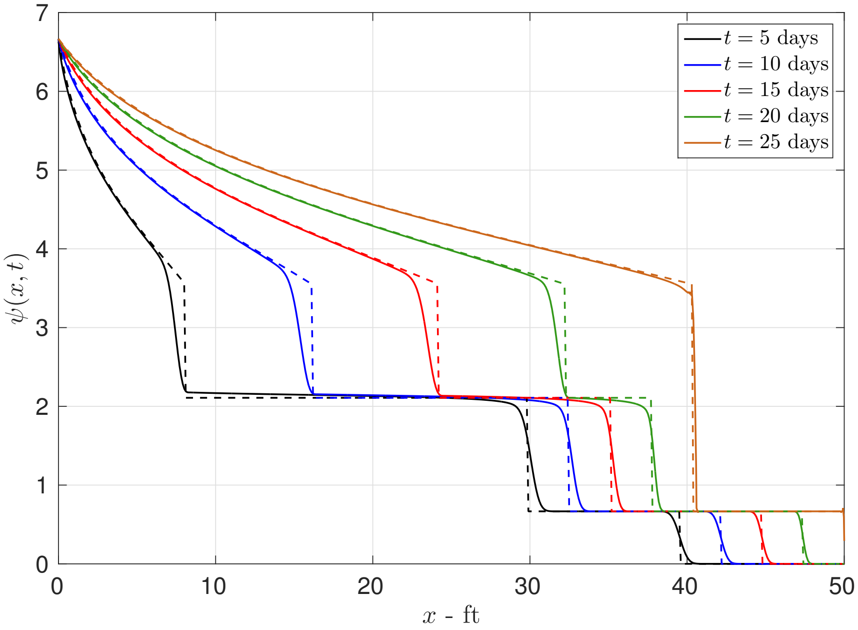


Figure 3-7: Comparison of space-time DG (solid lines) and exact (dashed lines) adjoint solutions at different times, for output J_T.

3.2.2 Output: volume integral over space-time domain

This section presents the adjoint problem and its solution for the Buckley-Leverett problem, with the output functional given in Eq. (3.59),

$$J = \int_\Omega g(S_w) d\Omega = \int_\Omega S_w^2(x,t) d\Omega. \tag{3.59}$$

Noting that J is exactly in the form of the general output function considered in Section 3.1.2, the results derived previously are applicable to this specific problem. Using Eq. (3.37), the adjoint equation for this problem is given by,

$$\phi \frac{\partial \psi}{\partial t} + \left(u_T \frac{\partial f_w}{\partial S_w} \right) \frac{\partial \psi}{\partial x} = -\frac{\partial g}{\partial S_w}. \tag{3.60}$$

As before, no adjoint BCs are required for the left and bottom boundaries, and the right boundary remains a homogeneous Dirichlet condition. The adjoint boundary condition at $t = T$ is exactly as given in Eq. (3.38),

$$\psi(x,T) = 0, \qquad \forall x \neq x_s(T). \tag{3.61}$$

The results given in Eq. (3.56) - (3.58) are valid for this output functional as well, showing that $\psi(x_s^-, t) = \psi_s(t)$, and that $\psi(x_s^+, t)$ and $\psi_s(t)$ may differ by an arbitrary amount.

The ODE governing $\psi_s(t)$, given by Eq. (3.39) - (3.40), simplifies to

$$\frac{d\psi_s}{dt} = -\frac{1}{\phi} \left(S_w(x_s^+, t) + S_w(x_s^-, t) \right), \tag{3.62}$$

subject to the condition,

$$\psi_s(T) = 0. \tag{3.63}$$

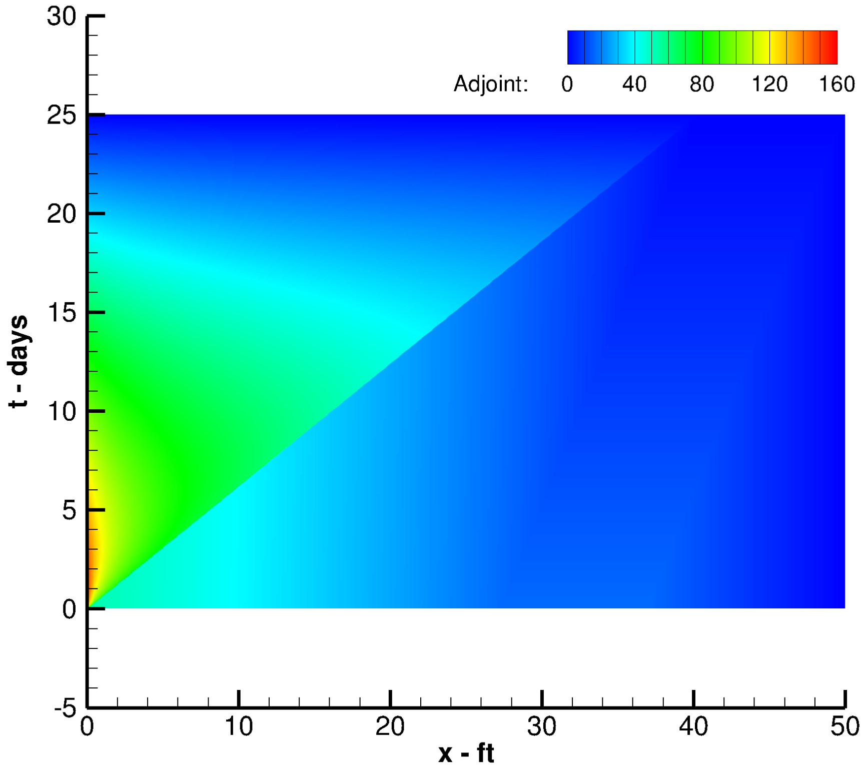


Figure 3-8: Exact adjoint solution for output J.

Noting that the exact solution of S_w to the left and right of the shock is constant in time, and solving the ODE given by Eq. (3.62) - (3.63) yields the following expression for $\psi_s(t)$,

$$\psi_s(t) = \frac{1}{\phi}\Big(S_w(x_s^+, t) + S_w(x_s^-, t)\Big)(T - t) \tag{3.64}$$

$$= \frac{1}{36}(5\sqrt{249} - 3)(T - t). \tag{3.65}$$

Figure 3-8 shows a contour plot of the analytical adjoint solution in the space-time domain (obtained using the approach outlined in Appendix C of [80]). Figure 3-9 contains the same plot for the numerical adjoint solution, obtained by a second-order space-time DG finite element method on a structured triangular mesh with 750, 000 degrees of freedom. The source term in Eq. (3.60) causes the adjoint to increase along each characteristic path emanating from the shock, or the top and right boundaries. Figure 3-10 compares the DG adjoint solutions (solid lines) at different times, with the

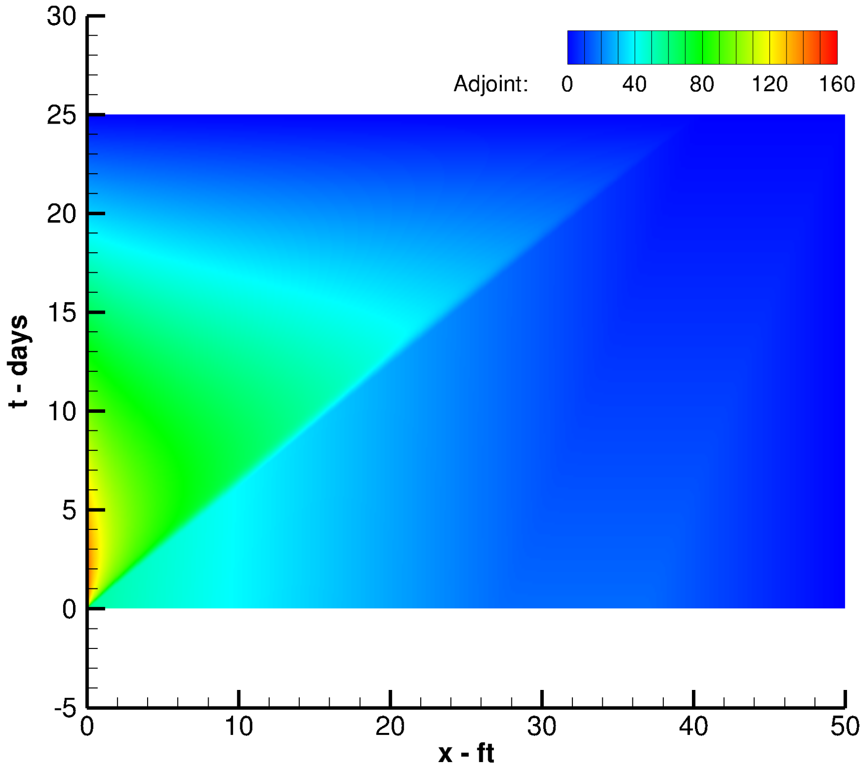


Figure 3-9: Numerical adjoint solution for output J from a second-order space-time DG method with 750,000 DOF.

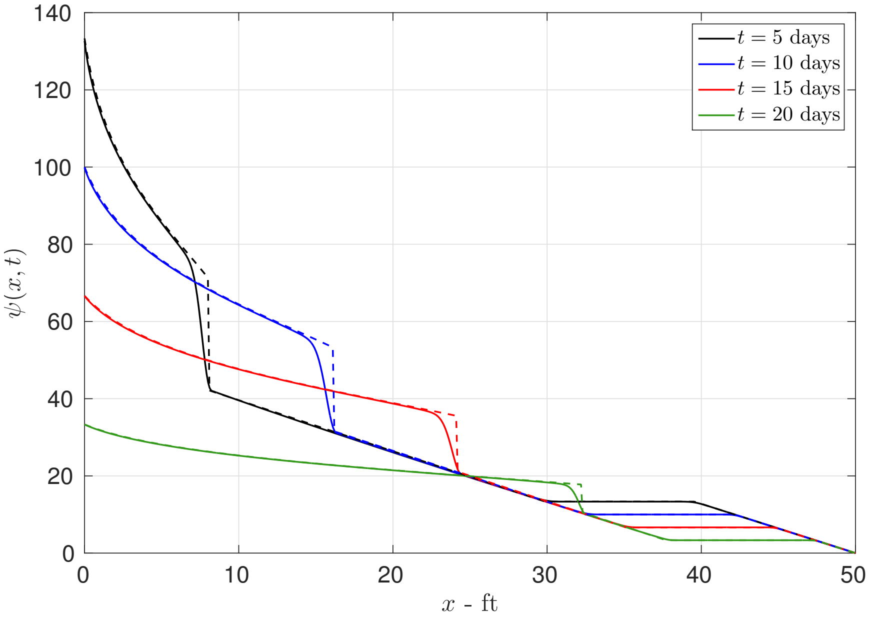


Figure 3-10: Comparison of space-time DG (solid lines) and exact (dashed lines) adjoint solutions at different times, for output J.

corresponding exact solutions (dashed lines). The space-time DG solutions agree well with the analytical results in general, except in the vicinity of solution discontinuities.

3.3 Two-phase flow equations

This section presents a derivation of the adjoint equations for the 1D compressible two-phase flow equations in mass conservation form. The wetting phase pressure $p(x,t)$ and the wetting phase saturation $S(x,t)$ are chosen as the dependent states. The governing equations for the wetting (w) and non-wetting (n) phases are given by,

$$\begin{aligned} (\rho_w \phi S)_t \qquad &- (\rho_w K \lambda_w p_x)_x &&= \rho_w q_w, && (3.66)\\ (\rho_n \phi (1-S))_t &- (\rho_n K \lambda_n (p_x + p_c' S_x))_x &&= \rho_n q_n, && (3.67) \end{aligned}$$

where $\rho_w(p)$ and $\rho_n(p)$ are the phase densities, $\lambda_w(S)$ and $\lambda_n(S)$ are the relative phase mobilities, K is the rock permeability, $\phi(p)$ is the rock porosity, $p_c(S)$ is the capillary pressure, and $q_w(p,S)$ and $q_n(p,S)$ are source/sink terms for each phase. All spatial and temporal derivatives are denoted with $(\cdot)_x$ and $(\cdot)_t$ subscripts respectively, while all derivatives with respect to the state variables are denoted with primes (i.e. $\frac{\partial \rho_w}{\partial p} = \rho_w'$ and $\frac{\partial \lambda_w}{\partial S} = \lambda_w'$). Furthermore, $\frac{\partial p_c}{\partial S}$ is replaced with $\gamma(S)$ for the rest of this section.

Eqs. (3.66) and (3.67) are written in the space-time formulation as,

$$\nabla \cdot \vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) - \mathbf{Q}(\mathbf{u}) = 0, \qquad \vec{x} \in \Omega, \tag{3.68}$$

where $\mathbf{u} = (p, S)^T$,

$$\vec{\mathbf{F}}(\mathbf{u}, \nabla \mathbf{u}) = \begin{pmatrix} \vec{F}_w \\ \vec{F}_n \end{pmatrix} \tag{3.69}$$

$$= \left[\begin{pmatrix} -\rho_w K \lambda_w p_x \\ -\rho_n K \lambda_n (p_x + \gamma S_x) \end{pmatrix}, \begin{pmatrix} \rho_w \phi S \\ \rho_n \phi (1 - S) \end{pmatrix} \right],$$

$$\mathbf{Q}(\mathbf{u}) = \begin{pmatrix} Q_w \\ Q_n \end{pmatrix} = \begin{pmatrix} \rho_w q_w \\ \rho_n q_n \end{pmatrix}, \tag{3.70}$$

and $\Omega = [0, L] \times [0, T]$ is the space-time domain as before.

3.3.1 Output: volume integral over space-time domain

The adjoint analysis of the two-phase flow equations assumes the following volume integrated output functional,

$$J = \int_\Omega g(\mathbf{u}) \; d\Omega. \tag{3.71}$$

The extension of this analysis to boundary integral outputs follows the procedure described in Section 3.1.1 for the Buckley-Leverett equation. As before, the adjoint derivation considers infinitesimal perturbations to the solution, $\delta\mathbf{u}$, and equates the linearized weak form to the linearized output,

$$\int_\Omega \boldsymbol{\psi}^T \left(\nabla \cdot \delta\vec{\mathbf{F}} - \delta\mathbf{Q} \right) d\Omega = \int_\Omega \delta g \; d\Omega, \tag{3.72}$$

where the adjoint vector $\boldsymbol{\psi} = (\psi_w, \psi_n)^T$ contains the adjoint solutions for the wetting and non-wetting phase equations respectively. Expanding out the terms of each phase equation yields,

$$\int_\Omega \psi_w \left(\nabla \cdot \delta\vec{F}_w - \delta Q_w \right) d\Omega + \int_\Omega \psi_n \left(\nabla \cdot \delta\vec{F}_n - \delta Q_n \right) d\Omega = \int_\Omega \delta g \; d\Omega. \tag{3.73}$$

Performing integration by parts and substituting in the flux definitions, the integrand of the volume integral is given by,

$$
\begin{aligned}
&-(\psi_w)_t\,\delta(\rho_w\phi S) \\
&+(\psi_w)_x\,\delta(\rho_w K\lambda_w p_x) \\
&-(\psi_n)_t\,\delta(\rho_n\phi(1-S)) \\
&+(\psi_n)_x\,\delta(\rho_n K\lambda_n(p_x+\gamma S_x)) \\
&-\psi_w\,\delta(\rho_w q_w)-\psi_n\,\delta(\rho_n q_n) \quad =\delta g.
\end{aligned}
\tag{3.74}
$$

Further use of integration by parts and the chain rule produces the following form of the volume integrand where only variations of p and S appear,

$$
\begin{aligned}
&-(\psi_w)_t\,[(\rho_w'\phi+\rho_w\phi')S\delta p && +\rho_w\phi\delta S] \\
&-(\psi_n)_t\,[(\rho_n'\phi+\rho_n\phi')(1-S)\delta p && -\rho_n\phi\delta S] \\
&+(\psi_w)_x\,[\rho_w'K\lambda_w p_x\delta p && +\rho_w K\lambda_w' p_x\delta S] \\
&+(\psi_n)_x\,[(\rho_n'K\lambda_n)(p_x+\gamma S_x)\delta p] && \\
&+(\psi_n)_x\,[(\rho_n K\lambda_n')(p_x+\gamma S_x)\delta S] && \\
&+(\psi_n)_x\,[\rho_n K\lambda_n\gamma' S_x\delta S] && \\
&-\psi_w\,[(\rho_w' q_w+\rho_w q_{w_p})\delta p && +\rho_w q_{w_S}\delta S] \\
&-\psi_n\,[(\rho_n' q_n+\rho_n q_{n_p})\delta p && +\rho_n q_{n_S}\delta S] \\
&-[\rho_w K\lambda_w(\psi_w)_x]_x\,\delta p && \\
&-[\rho_n K\lambda_n(\psi_n)_x]_x\,\delta p && -[\rho_n K\lambda_n\gamma(\psi_n)_x]_x\,\delta S=g_p\delta p+g_S\delta S,
\end{aligned}
\tag{3.75}
$$

where $q_{w_p}=\frac{\partial q_w}{\partial p}, q_{w_S}=\frac{\partial q_w}{\partial S}, q_{n_p}=\frac{\partial q_n}{\partial p}, q_{n_S}=\frac{\partial q_n}{\partial S}, g_p=\frac{\partial g}{\partial p}$ and $g_S=\frac{\partial g}{\partial S}$.

Grouping together terms that multiply δp and noting that Eq. (3.75) holds for any δp, yields the first adjoint equation, given in Eq. (3.76). Repeating the process for terms multiplying δS yields the second adjoint equation, given in Eq. (3.77).

$$\begin{aligned}
&- (\psi_w)_t \cdot (\rho_w'\phi + \rho_w\phi')S \\
&- (\psi_n)_t \cdot (\rho_n'\phi + \rho_n\phi')(1 - S) \\
&+ (\psi_w)_x \cdot (\rho_w' K\lambda_w p_x) \\
&+ (\psi_n)_x \cdot [(\rho_n' K\lambda_n)(p_x + \gamma S_x)] \\
&- [\rho_w K\lambda_w(\psi_w)_x]_x - [\rho_n K\lambda_n(\psi_n)_x]_x \\
&- \psi_w \cdot (\rho_w' q_w + \rho_w q_{w_p}) - \psi_n \cdot (\rho_n' q_n + \rho_n q_{n_p}) = g_p
\end{aligned} \tag{3.76}$$

$$\begin{aligned}
&- (\psi_w)_t \cdot (\rho_w\phi) \\
&- (\psi_n)_t \cdot (-\rho_n\phi) \\
&+ (\psi_w)_x \cdot (\rho_w K\lambda_w' p_x) \\
&+ (\psi_n)_x \cdot [(\rho_n K\lambda_n')(p_x + \gamma S_x) + \rho_n K\lambda_n\gamma' S_x] \\
&- [\rho_n K\lambda_n\gamma(\psi_n)_x]_x \\
&- \psi_w \cdot (\rho_w q_{w_S}) - \psi_n \cdot (\rho_n q_{n_S}) \qquad\qquad = g_S
\end{aligned} \tag{3.77}$$

Next, the boundary conditions of the adjoint problem are derived by collecting the boundary integral terms from Eq. (3.72), and accounting for the integration by parts that led to Eq. (3.76) and (3.77). Specifically, the boundary integrals at $t = T$ are

$$\begin{aligned}
&\int_0^L [\psi_w(x,T)(\rho_w'\phi + \rho_w\phi')S]\, \delta p \; dx \\
&+ \int_0^L [\psi_n(x,T)(\rho_n'\phi + \rho_n\phi')(1 - S)]\, \delta p \; dx \\
&+ \int_0^L [\psi_w(x,T)(\rho_w\phi) + \psi_n(x,T)(-\rho_n\phi)]\, \delta S \; dx = 0.
\end{aligned} \tag{3.78}$$

Requiring Eq. (3.78) to hold for any δp and δS gives the following conditions on the adjoint variables,

$$\psi_w(x,T)(\rho_w'\phi + \rho_w\phi')S + \psi_n(x,T)(\rho_n'\phi + \rho_n\phi')(1 - S) = 0, \tag{3.79}$$

and,

$$\rho_w \psi_w(x,T) - \rho_n \psi_n(x,T) = 0. \tag{3.80}$$

Similarly, isolating the boundary integrals for the right boundary gives,

$$\begin{aligned}
&\int_0^T \left[-\rho_w' K \lambda_w p_x \psi_w - \rho_n' K \lambda_n (p_x + \gamma S_x)\psi_n\right] \delta p \\
&+ \int_0^T \left[\rho_w K \lambda_w (\psi_w)_x + \rho_n K \lambda_n (\psi_n)_x\right] \delta p \\
&+ \int_0^T \left[-\rho_w K \lambda_w' p_x \psi_w\right] \delta S \\
&+ \int_0^T \left[-(\rho_n K \lambda_n' (p_x + \gamma S_x) + \rho_n K \lambda_n \gamma' S_x)\psi_n\right] \delta S \\
&+ \int_0^T \left[\rho_n K \lambda_n \gamma (\psi_n)_x\right] \delta S \\
&+ \int_0^T \left[-\rho_w K \lambda_w \psi_w - \rho_n K \lambda_n \psi_n\right] \delta p_x \\
&+ \int_0^T \left[-\rho_n K \lambda_n \gamma \psi_n\right] \delta S_x = 0.
\end{aligned} \tag{3.81}$$

Inspecting the integrands in Eq. (3.81), and accounting for the nature of the imposed primal boundary conditions yields the adjoint boundary conditions at the right boundary. For example, if the primal problem imposes Dirichlet boundary conditions for pressure and saturation at the right boundary, then $\delta p(L,t) = \delta S(L,t) = 0$, and therefore the adjoint solutions would only need to satisfy the conditions corresponding to the δp_x and δS_x terms. Specifically,

$$\rho_w K \lambda_w \psi_w(L,t) + \rho_n K \lambda_n \psi_n(L,t) = 0, \tag{3.82}$$

$$\rho_n K \lambda_n \gamma \psi_n(L,t) = 0. \tag{3.83}$$

Assuming $\gamma \neq 0$, the two conditions above reduce to $\psi_w(L,t) = \psi_n(L,t) = 0$. Isolating the boundary integrals for the left boundary produces an equation similar to Eq. (3.81), from which the adjoint boundary conditions can be determined in an analogous manner to the right boundary. As before, the primal initial condition eliminates the need for an adjoint boundary condition at $t = 0$.

3.3.2 Relationship with Buckley-Leverett

It is possible to reduce the two-phase flow equations presented in Eq. (3.66) and (3.67) to the Buckley-Leverett equation given in Eq. (3.41) by assuming incompressibility (i.e. $\rho_w' = \rho_n' = \phi' = 0$), zero capillary pressure (i.e. $\gamma = 0$), and the absence of source terms ($q_w = q_n = 0$). Under these assumptions, the primal equations in Eq. (3.66) and (3.67) reduce to,

$$\phi S_t - \left(K\lambda_w p_x\right)_x = 0, \tag{3.84}$$

$$-\phi S_t - \left(K\lambda_n p_x\right)_x = 0. \tag{3.85}$$

Taking the sum of Eq. (3.84) and (3.85) produces an elliptic pressure equation,

$$-\left(K(\lambda_w + \lambda_n)p_x\right)_x = 0. \tag{3.86}$$

The Buckley-Leverett equation is a combination of the wetting-phase saturation equation (Eq. (3.84)) and the pressure equation (Eq. (3.86)). Integrating Eq. (3.86) in space shows that $-K(\lambda_w + \lambda_n)p_x$ is equal to a constant (namely, the total velocity u_T), thereby allowing the spatial flux in Eq. (3.84) to be written as,

$$-K\lambda_w p_x = u_T \frac{\lambda_w}{\lambda_w + \lambda_n} = u_T f_w(S), \tag{3.87}$$

where the last equality uses the definition of the wetting phase fractional flow function, $f_w(S) = \frac{\lambda_w}{\lambda_w + \lambda_n}$. Using Eq. (3.87) in Eq. (3.84) yields the Buckley-Leverett equation given in Eq. (3.41).

Eq. (3.84) and (3.86) are written in the space-time formulation as,

$$\nabla \cdot \vec{\hat{\mathbf{F}}}(\hat{\mathbf{u}}, \nabla\hat{\mathbf{u}}) = 0, \tag{3.88}$$

where $\hat{\mathbf{u}} = (p, S)^T$, and,

$$\vec{\hat{\mathbf{F}}} = \begin{pmatrix} \vec{\hat{F}}_{BL} \\ \\ \vec{\hat{F}}_{p} \end{pmatrix} = \left[\begin{pmatrix} -K\lambda_w p_x \\ \\ -K(\lambda_w + \lambda_n) p_x \end{pmatrix}, \begin{pmatrix} \phi S \\ \\ 0 \end{pmatrix} \right]. \tag{3.89}$$

As before, the adjoint problem for this new, but equivalent, set of primal equations is obtained by equating the linearized weak form to the linearized output,

$$\int_\Omega \psi_{BL} \left(\nabla \cdot \delta \vec{\hat{F}}_{BL} \right) d\Omega + \int_\Omega \psi_p \left(\nabla \cdot \delta \vec{\hat{F}}_{p} \right) d\Omega = \int_\Omega \delta g \; d\Omega, \tag{3.90}$$

where the new adjoint vector $\hat{\boldsymbol{\psi}} = [\psi_{BL}, \psi_p]^T$ contains the adjoint solutions for the Buckley-Leverett and pressure equations respectively.

The relationship between $\hat{\boldsymbol{\psi}}$ and $\boldsymbol{\psi}$ is obtained by via the analysis presented in Appendix D, which derives a simple relationship between the adjoint solutions of two equivalent sets of primal equations that are linear combinations of each other. Following the definitions given in Appendix D, the transformation matrix $\mathbf{H}$ from the wetting-nonwetting primal equations to the Buckley-Leverett-pressure primal equations is,

$$\mathbf{H} = \begin{pmatrix} \frac{1}{\rho_w} & 0 \\ \\ \frac{1}{\rho_w} & \frac{1}{\rho_n} \end{pmatrix}. \tag{3.91}$$

Eq. (D.16) states that $\hat{\boldsymbol{\psi}} = \mathbf{H}^{-T} \boldsymbol{\psi}$, which when applied to this particular problem gives,

$$\begin{pmatrix} \psi_{BL} \\ \\ \psi_p \end{pmatrix} = \begin{pmatrix} \rho_w \psi_w - \rho_n \psi_n \\ \\ \rho_n \psi_n \end{pmatrix}. \tag{3.92}$$

The ability to derive analytical solutions for ψ_{BL} makes the above relationship useful for verifying numerical adjoint solutions of the two-phase flow equations, which do not have analytical solutions in general.

3.3.3 Numerical results

The space-time DG finite element method described in Chapter 2 is used to compute the adjoint solutions of a two-phase flow problem that is consistent with the Buckley-Leverett problem defined in Eq. (3.41) - (3.43). This requires setting Dirichlet BCs for saturation S along the $t = 0$, $x = 0$ and $x = L$ boundaries. The pressure p requires a Neumann BC at the $x = 0$ boundary, and Dirichlet BCs at the $t = 0$ and $x = L$ boundaries. The pressure gradient used for the Neumann BC is calculated from Eq. (3.87), to be consistent with the Dirichlet saturation condition given by Eq. (3.43) on the $x = 0$ boundary. No boundary conditions are imposed at the $t = T$ boundary, where all fluxes are evaluated from the states in the interior of the domain. The problem is incompressible and contains no source terms. However, a small amount of capillary pressure ($\gamma = 0.1$) is required to stabilize oscillations that occur at the shock due to the Gibbs' phenomenon. Although this is a slight deviation from the Buckley-Leverett problem, which assumes zero capillary effects, it has no discernible impact on the numerical solutions.

Figures 3-11 and 3-12 show contour plots of the two-phase flow adjoint solutions, ψ_w and ψ_n respectively, obtained using a second-order space-time DG finite element method with approximately 750,000 degrees of freedom per state variable. Figure 3-13 shows a plot of ψ_{BL}, computed using ψ_w and ψ_n according to the first equation in Eq. (3.92). Visually comparing Figure 3-13 with Figure 3-8 demonstrates that ψ_{BL} agrees well with the adjoint solution of the Buckley-Leverett equation. However, in order to make a more formal comparison, profiles of ψ_{BL} at different times are compared with the analytical Buckley-Leverett adjoint derived in Section 3.2.2, as shown in Figure 3-14. The near-perfect agreement between the two solutions except in the vicinity of the shocks provides a satisfactory numerical confirmation of Eq. (3.92).

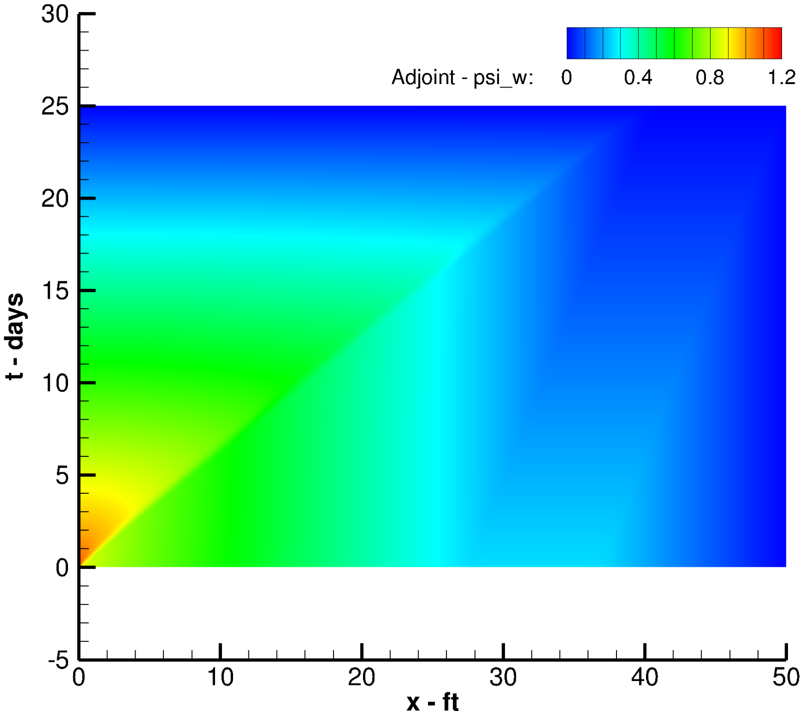


Figure 3-11: Numerical adjoint solution ψ_w for output J from a second-order space-time DG method with 750,000 DOF per state variable.

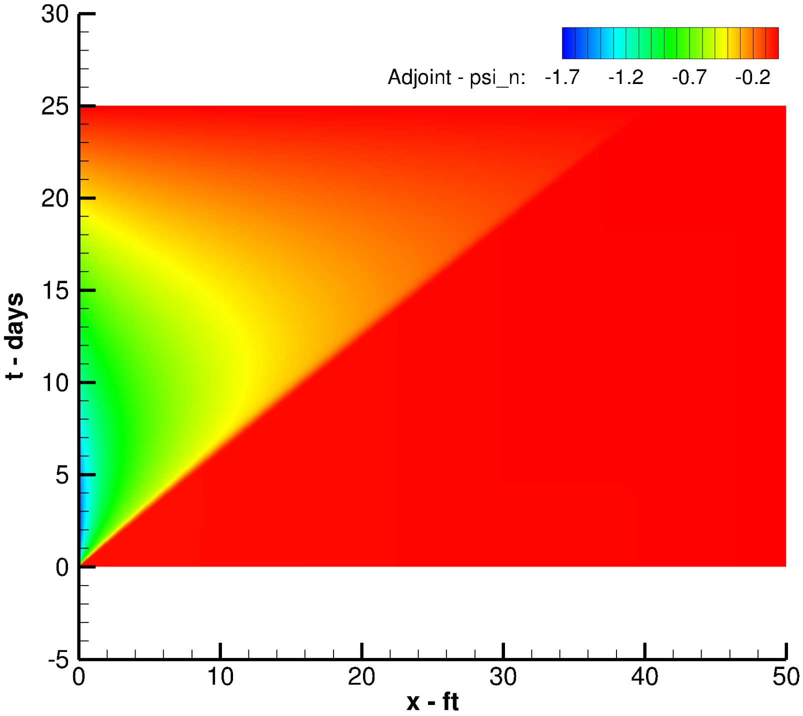


Figure 3-12: Numerical adjoint solution ψ_n for output J from a second-order space-time DG method with 750,000 DOF per state variable.

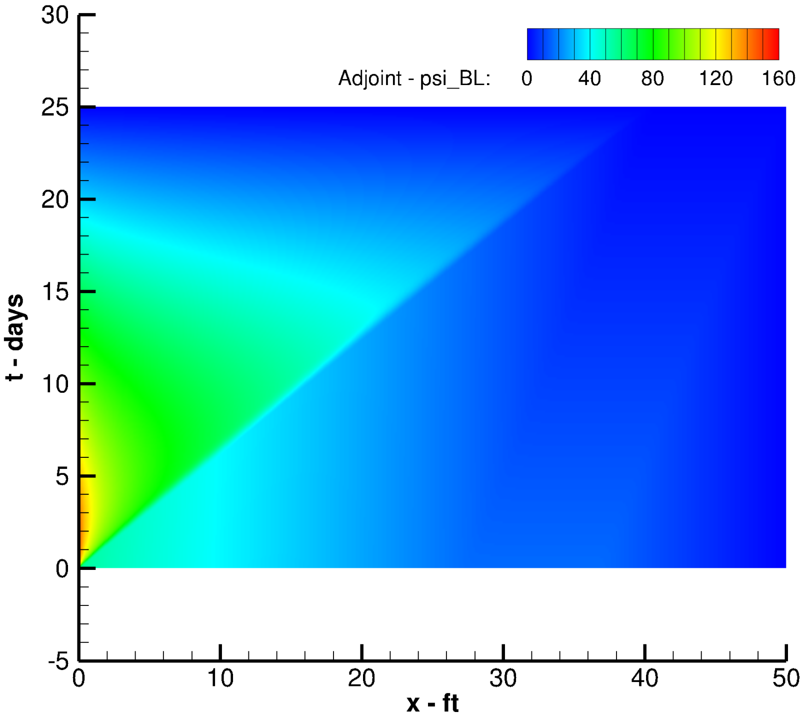


Figure 3-13: Space-time contour plot of $\psi_{BL} = \rho_w \psi_w - \rho_n \psi_n$, computed from the numerical adjoint solutions.

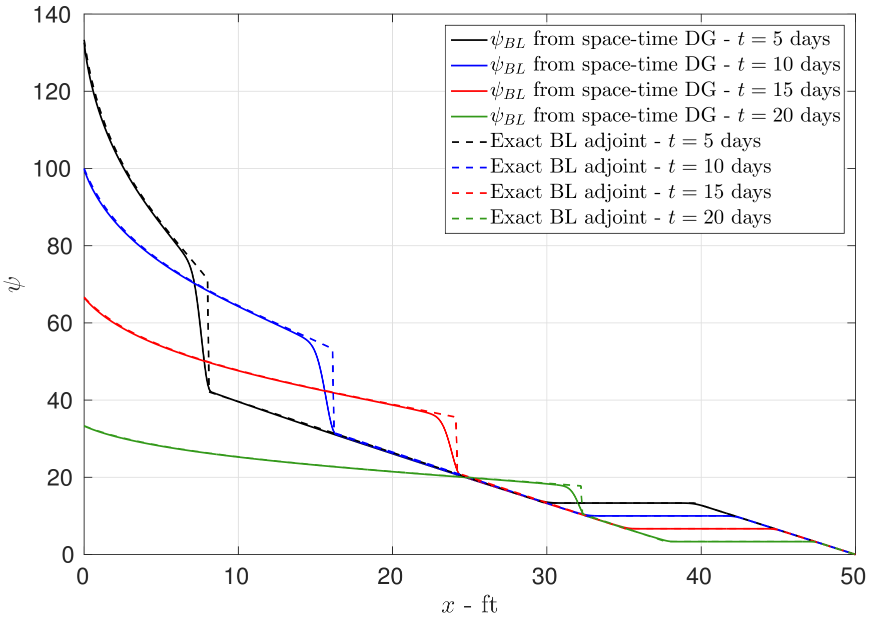


Figure 3-14: Comparison of ψ_{BL} with the exact Buckley-Leverett adjoint at different times.

3.4 Summary

This chapter presented a derivation of the adjoint equation and boundary conditions for a scalar conservation law containing a shock, for two different output functionals: one involving a spatial integral and the other involving a space-time integral of solution dependent quantities. The results were specialized to the Buckley-Leverett problem, where attention to the combined rarefaction-shock wave behavior of the equations was essential to produce the correct analytical solution. In contrast to the behavior of equations with convex flux functions, such as the Burgers' equation, where the adjoint is continuous across a shock, the Buckley-Leverett equation is found to admit a discontinuous jump in adjoint value across a shock.

The adjoint equations for the compressible two-phase flow equations in mass conservation form were also presented, including a relationship between the adjoint solutions of the two-phase flow and Buckley-Leverett problems under appropriate assumptions. Numerical results from the space-time DG method were observed to be in good agreement with the derived analytical solutions.

Chapter 4

Upwinding the two-phase flow equations

It is well known that the incompressible two-phase flow equations possess a hyperbolic-elliptic nature due to the presence of an underlying elliptic pressure equation and a hyperbolic (or near-hyperbolic) saturation transport equation. Under assumptions of zero capillary pressure and one dimensional flow, this saturation equation reduces to the well-known Buckley-Leverett equation. This relationship was used in the previous chapter to derive a relationship between the adjoint solutions of the two-phase flow and Buckley-Leverett equations. However, if the two-phase flow equations are expressed in mass conservation form, as is done in most industrial practices, the equations "appear" as a pair of coupled parabolic equations, effectively concealing their hyperbolic nature. Consider the two-phase flow equations given in mass conservation form below for the wetting (w) and non-wetting (n) phases, with the primary unknowns being the non-wetting phase pressure p_n and the wetting-phase saturation S_w,

$$\frac{\partial}{\partial t}\left(\rho_w \phi S_w\right) - \nabla \cdot \left(\rho_w \lambda_w \mathbf{K} \nabla p_n - \rho_w \lambda_w \mathbf{K} p_{c_S} \nabla S_w\right) = 0, \tag{4.1}$$

$$\frac{\partial}{\partial t}\left(\rho_n \phi (1 - S_w)\right) - \nabla \cdot \left(\rho_n \lambda_n \mathbf{K} \nabla p_n\right) = 0, \tag{4.2}$$

where ρ_w, ρ_n are the phase densities, $\lambda_w(S_w) = \frac{k_{rw}}{\mu_w}, \lambda_n(S_w) = \frac{k_{rn}}{\mu_n}$ are the relative phase mobilities, ϕ is the rock porosity, $\mathbf{K}$ is the absolute rock permeability tensor, and $p_{c_S} = \frac{dp_c}{dS_w}$ where $p_c(S_w)$ is the capillary pressure. The work in this thesis neglects gravitational effects, but they can be easily incorporated if needed by replacing the phase pressures with phase potentials that include the hydrostatic pressure. If the spatial fluxes inside the divergence operators in Eqs. 4.1 - 4.2 are interpreted as diffusive fluxes and discretized using a standard DG method such as the BR2 method given in Eq. 2.13, it results in a centered discretization of the spatial operators. However, a centered discretization of an advection-dominant problem is known to cause oscillations and instabilities. This therefore suggests that the DG discretization of the spatial operators in Eqs. 4.1 - 4.2 requires some modification, typically an upwinding bias, in order to be stable in the advection-dominant limit.

This chapter presents the derivation of such a modification to the BR2 scheme, obtained via a linearized analysis of the two-phase flow equations in both continuous and discrete settings. The various steps of the analysis are summarized in Figure 4-1, where the blue blocks represent the continuous equations, yellow blocks represent the standard DG weak form equations that result in centered discretizations, and the green blocks represent modified DG weak form equations where the saturation behavior has been correctly upwinded. The analysis first linearizes the continuous two-phase flow equations in mass conservation form and then manipulates them into a linearized pressure-saturation form (bottom-left block). This is then discretized using the DG method to obtain a centered discretization for the linearized pressure equation, and an appropriately upwinded discretization for the linearized saturation equation (bottom-right block). The desired modification terms to the DG weak form (dashed arrow) are designed such that a similar linearization of the final modified DG discretization (top-right block) would produce the same discrete linearized pressure-saturation form.

Although the analysis is performed using the BR2 scheme, it is expected that the resulting modifications to the discretization will be equally applicable to other centered DG diffusive flux discretizations such as the local discontinuous Galerkin method (LDG) [42], compact discontinuous Galerkin method (CDG) [117] and the interior penalty Galerkin methods (SIPG, IIPG).

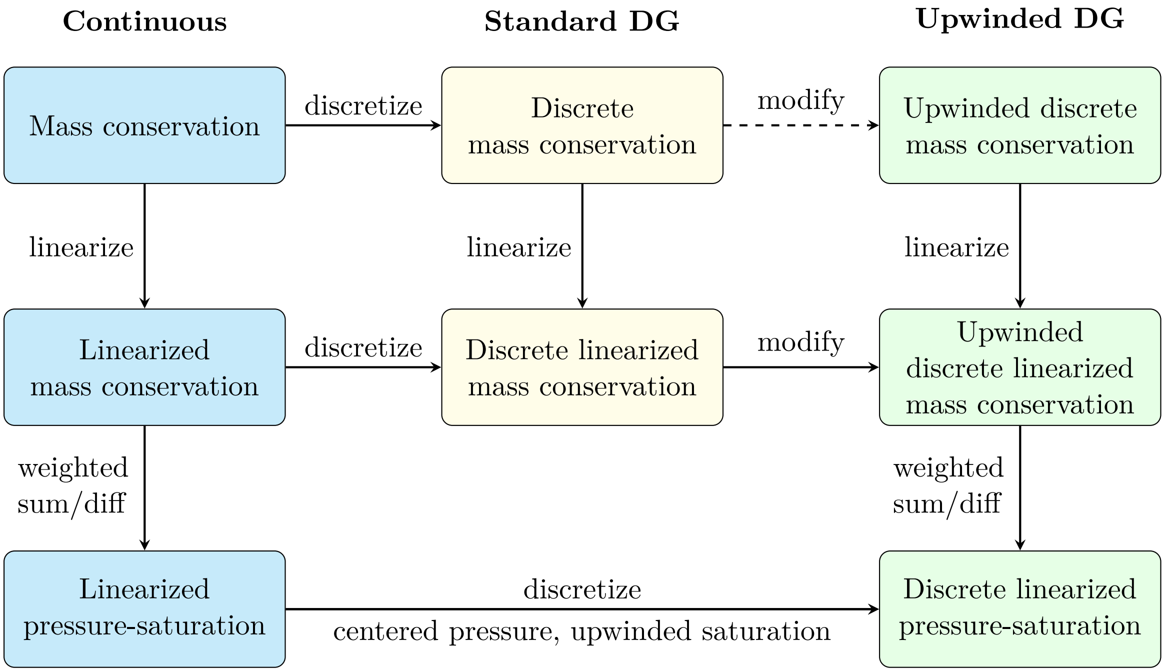

Continuous
Standard DG
Upwinded DG
Mass conservation
discretize
Discrete mass conservation
modify
Upwinded discrete mass conservation
linearize
linearize
linearize
Linearized mass conservation
discretize
Discrete linearized mass conservation
modify
Upwinded discrete linearized mass conservation
weighted sum/diff
weighted sum/diff
Linearized pressure-saturation
discretize
centered pressure, upwinded saturation
Discrete linearized pressure-saturation

Figure 4-1: Outline of linearized analysis for deriving upwinding modifications

4.1 Continuous linearized analysis

Consider a solution that is expressed as a perturbation about some mean pressure and saturation distribution,

$$p_n(\vec{x},t) = \bar{p}_n(\vec{x},t) + p_n'(\vec{x},t), \tag{4.3}$$

$$S_w(\vec{x},t) = \bar{S}_w(\vec{x},t) + S_w'(\vec{x},t), \tag{4.4}$$

where the mean solutions $\bar{p}_n$ and $\bar{S}_w$ satisfy Eqs. 4.1 - 4.2. Substituting the above perturbed solutions into Eqs. 4.1 - 4.2, canceling out the mean flow terms, and ignoring products of perturbations yields the linearized incompressible two-phase flow equations,

$$\rho_w \phi \frac{\partial S'_w}{\partial t} + \nabla \cdot \left(-\rho_w \bar{\lambda}_{w_S} \mathbf{K} \nabla \bar{p}_n \; S'_w + \rho_w \left(\bar{\lambda}_{w_S} \bar{p}_{c_S} + \bar{\lambda}_w \bar{p}_{c_{SS}} \right) \mathbf{K} \nabla \bar{S}_w \; S'_w \right) \tag{4.5}$$

$$- \nabla \cdot \left(\rho_w \bar{\lambda}_w \mathbf{K} \nabla p'_n - \rho_w \bar{\lambda}_w \mathbf{K} \bar{p}_{c_S} \nabla S'_w \right) = 0,$$

$$-\rho_n \phi \frac{\partial S'_w}{\partial t} + \nabla \cdot \left(-\rho_n \bar{\lambda}_{n_S} \mathbf{K} \nabla \bar{p}_n \; S'_w \right) - \nabla \cdot \left(\rho_n \bar{\lambda}_n \mathbf{K} \nabla p'_n \right) = 0, \tag{4.6}$$

where $\bar{\lambda}_{\alpha_S} = \frac{d\lambda_\alpha}{dS_w}\Big|_{\bar{S}_w}$, $\bar{p}_{c_S} = \frac{dp_c}{dS_w}\Big|_{\bar{S}_w}$, and $\bar{p}_{c_{SS}} = \frac{d^2 p_c}{dS_w^2}\Big|_{\bar{S}_w}$.

Taking the weighted sum of the linearized equations, $\rho_n \times$(Eq. 4.5)$+\rho_w \times$(Eq. 4.6), yields an elliptic "pressure" equation,

$$\nabla \cdot \left(-\rho_w \rho_n \left(\bar{\lambda}_{w_S} + \bar{\lambda}_{n_S} \right) \mathbf{K} \nabla \bar{p}_n \; S'_w + \rho_w \rho_n \left(\bar{\lambda}_{w_S} \bar{p}_{c_S} + \bar{\lambda}_w \bar{p}_{c_{SS}} \right) \mathbf{K} \nabla \bar{S}_w \; S'_w \right)$$

$$- \nabla \cdot \left(\rho_w \rho_n \left(\bar{\lambda}_w + \bar{\lambda}_n \right) \mathbf{K} \nabla p'_n - \rho_w \rho_n \bar{\lambda}_w \mathbf{K} \bar{p}_{c_S} \nabla S'_w \right) = 0. \tag{4.7}$$

Similarly, taking the weighted difference of the linearized equations, $\rho_n \bar{\lambda}_n \times$(Eq. 4.5)$-$ $\rho_w \bar{\lambda}_w \times$ (Eq. 4.6), yields a parabolic "saturation" equation,

$$\rho_w \rho_n \left(\bar{\lambda}_w + \bar{\lambda}_n \right) \phi \frac{\partial S'_w}{\partial t}$$

$$+ \nabla \cdot \left(-\rho_w \rho_n \left(\left(\bar{\lambda}_{w_S} \bar{\lambda}_n - \bar{\lambda}_w \bar{\lambda}_{n_S} \right) \mathbf{K} \nabla \bar{p}_n - \left(\bar{\lambda}_{w_S} \bar{\lambda}_n \bar{p}_{c_S} + \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_{SS}} \right) \mathbf{K} \nabla \bar{S}_w \right) \; S'_w \right)$$

$$- \nabla \cdot \left(-\rho_w \rho_n \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_S} \mathbf{K} \nabla S'_w \right) = 0. \tag{4.8}$$

The saturation equation above may be written in the form of the unsteady advection-diffusion PDE,

$$\frac{\partial S'_w}{\partial t} + \nabla \cdot \left(\bar{\vec{V}} S'_w - \bar{\mathbf{D}} \nabla S'_w \right) = 0, \tag{4.9}$$

where the advection velocity $\bar{\vec{V}}$ and the diffusion coefficient $\bar{\mathbf{D}}$ are given by,

$$\bar{\vec{V}} = -\frac{\bar{\lambda}_{w_S} \bar{\lambda}_n - \bar{\lambda}_w \bar{\lambda}_{n_S}}{\phi(\bar{\lambda}_w + \bar{\lambda}_n)} \mathbf{K} \nabla \bar{p}_n + \frac{\bar{\lambda}_{w_S} \bar{\lambda}_n \bar{p}_{c_S} + \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_{SS}}}{\phi(\bar{\lambda}_w + \bar{\lambda}_n)} \mathbf{K} \nabla \bar{S}_w, \tag{4.10}$$

$$\bar{\mathbf{D}} = \frac{-\bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_S} \mathbf{K}}{\phi(\bar{\lambda}_w + \bar{\lambda}_n)}. \tag{4.11}$$

Note that the saturation equation reduces to a purely hyperbolic PDE when there are no capillary effects ($p_c = 0$).

4.2 Discrete linearized analysis

Consider the following discontinuous Galerkin weak form for the incompressible two-phase flow equations in Eqs. 4.1 - 4.2, with additional stabilization terms g_w and g_n that are yet to be determined. The DG method seeks a discrete solution $\mathbf{u}_{h,p} = [p_n, S_w] \in \mathcal{V}_{h,p}$ that satisfies,

$$\begin{aligned}
\sum_{\kappa \in \mathcal{T}_h} \int_\kappa v \rho_w \phi \frac{\partial S_w}{\partial t} \; d\Omega + \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla v \cdot \left(\rho_w \lambda_w \mathbf{K} \nabla p_n - \rho_w \lambda_w p_{c_S} \mathbf{K} \nabla S_w \right) \; d\Omega \\
- \sum_{f \in \Gamma_I} \int_f \llbracket v \rrbracket \cdot \left\{ \rho_w \lambda_w \mathbf{K} \left(\nabla p_n + \eta_f \vec{r}_p \left(\llbracket p_n \rrbracket \right) \right) \right\} \; d\Gamma \\
- \sum_{f \in \Gamma_I} \int_f \llbracket v \rrbracket \cdot \left\{ -\rho_w \lambda_w p_{c_S} \mathbf{K} \left(\nabla S_w + \eta_f \vec{r}_S \left(\llbracket S_w \rrbracket \right) \right) \right\} \; d\Gamma \\
- \sum_{f \in \Gamma_I} \int_f \left\{ \rho_w \lambda_w \mathbf{K}^T \nabla v \right\} \cdot \llbracket p_n \rrbracket \; d\Gamma - \sum_{f \in \Gamma_I} \int_f \left\{ -\rho_w \lambda_w p_{c_S} \mathbf{K}^T \nabla v \right\} \cdot \llbracket S_w \rrbracket \; d\Gamma \\
+ \sum_{f \in \Gamma_I} \int_f g_w(v^\pm, \mathbf{u}_{h,p}^\pm, \nabla \mathbf{u}_{h,p}^\pm; \vec{n}^+) \; d\Gamma = 0, \\
\forall v \in \mathcal{V}_{h,p},
\end{aligned} \tag{4.12}$$

$$\begin{aligned}
\sum_{\kappa \in \mathcal{T}_h} \int_\kappa -v \rho_n \phi \frac{\partial S_w}{\partial t} \; d\Omega + \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla v \cdot \left(\rho_n \lambda_n \mathbf{K} \nabla p_n \right) \; d\Omega \\
- \sum_{f \in \Gamma_I} \int_f \llbracket v \rrbracket \cdot \left\{ \rho_n \lambda_n \mathbf{K} \left(\nabla p_n + \eta_f \vec{r}_p \left(\llbracket p_n \rrbracket \right) \right) \right\} \; d\Gamma \\
- \sum_{f \in \Gamma_I} \int_f \left\{ \rho_n \lambda_n \mathbf{K}^T \nabla v \right\} \cdot \llbracket p_n \rrbracket \; d\Gamma \\
+ \sum_{f \in \Gamma_I} \int_f g_n(v^\pm, \mathbf{u}_{h,p}^\pm, \nabla \mathbf{u}_{h,p}^\pm; \vec{n}^+) \; d\Gamma = 0, \\
\forall v \in \mathcal{V}_{h,p}.
\end{aligned} \tag{4.13}$$

All of the spatial flux terms in Eqs. 4.12 - 4.13, except for the g_w and g_n terms, are obtained by expanding the BR2 operator given in Eq. 2.13 for the wetting and non-wetting equations separately, and ignoring the boundary terms for simplicity. The lifting operators for the primary variables p_n and S_w are represented by $\vec{r}_p$ and $\vec{r}_S$ respectively.

As done previously for the continuous case, the DG weak form given by Eqs. 4.12 - 4.13 is then linearized about the mean pressure and saturation solution, yielding the linearized weak form of the incompressible two-phase flow equations,

$$\begin{aligned}
&\sum_{\kappa\in\mathcal{T}_h}\int_\kappa v\rho_w\phi\frac{\partial S'_w}{\partial t}\;d\Omega+\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla v\cdot\left(\rho_w\bar{\lambda}_{w_S}\mathbf{K}\nabla\bar{p}_n S'_w+\rho_w\bar{\lambda}_w\mathbf{K}\nabla p'_n\right)\;d\Omega\\
&+\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla v\cdot\left(-\rho_w(\bar{\lambda}_{w_S}\bar{p}_{c_S}+\bar{\lambda}_w\bar{p}_{c_{SS}})\mathbf{K}\nabla\bar{S}_w S'_w-\rho_w\bar{\lambda}_w\bar{p}_{c_S}\mathbf{K}\nabla S'_w\right)\;d\Omega\\
&-\sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\left\{\rho_w\bar{\lambda}_{w_S}\mathbf{K}\nabla\bar{p}_n S'_w+\rho_w\bar{\lambda}_w\mathbf{K}\left(\nabla p'_n+\eta_f\vec{r}'_p\left(\llbracket p'_n\rrbracket\right)\right)\right\}\;d\Gamma\\
&-\sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\left\{-\rho_w(\bar{\lambda}_{w_S}\bar{p}_{c_S}+\bar{\lambda}_w\bar{p}_{c_{SS}})\mathbf{K}\nabla\bar{S}_w S'_w\right\}\;d\Gamma\\
&-\sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\left\{-\rho_w\bar{\lambda}_w\bar{p}_{c_S}\mathbf{K}\left(\nabla S'_w+\eta_f\vec{r}'_S\left(\llbracket S'_w\rrbracket\right)\right)\right\}\;d\Gamma\\
&-\sum_{f\in\Gamma_I}\int_f \left\{\rho_w\bar{\lambda}_w\mathbf{K}^T\nabla v\right\}\cdot\llbracket p'_n\rrbracket\;d\Gamma-\sum_{f\in\Gamma_I}\int_f \left\{-\rho_w\bar{\lambda}_w\bar{p}_{c_S}\mathbf{K}^T\nabla v\right\}\cdot\llbracket S'_w\rrbracket\;d\Gamma\\
&+\sum_{f\in\Gamma_I}\int_f g'_w\;d\Gamma=0,\quad\forall v\in\mathcal{V}_{h,p},
\end{aligned}\tag{4.14}$$

and,

$$\begin{aligned}
&\sum_{\kappa\in\mathcal{T}_h}\int_\kappa -v\rho_n\phi\frac{\partial S'_w}{\partial t}\;d\Omega+\sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla v\cdot\left(\rho_n\bar{\lambda}_{n_S}\mathbf{K}\nabla\bar{p}_n S'_w+\rho_n\bar{\lambda}_n\mathbf{K}\nabla p'_n\right)\;d\Omega\\
&-\sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\left\{\rho_n\bar{\lambda}_{n_S}\mathbf{K}\nabla\bar{p}_n S'_w+\rho_n\bar{\lambda}_n\mathbf{K}\left(\nabla p'_n+\eta_f\vec{r}'_p\left(\llbracket p'_n\rrbracket\right)\right)\right\}\;d\Gamma\\
&-\sum_{f\in\Gamma_I}\int_f \left\{\rho_n\bar{\lambda}_n\mathbf{K}^T\nabla v\right\}\cdot\llbracket p'_n\rrbracket\;d\Gamma\\
&+\sum_{f\in\Gamma_I}\int_f g'_n\;d\Gamma=0,\quad\forall v\in\mathcal{V}_{h,p},
\end{aligned}\tag{4.15}$$

where it is assumed that $[\![\bar{p}_n]\!] = [\![\bar{S}_w]\!] = \vec{r}_p([\![\bar{p}_n]\!]) = \vec{r}_S([\![\bar{S}_w]\!]) = 0$. Note that Eqs. 4.14 - 4.15 are also equivalent to the weak form obtained by discretizing the linearized two-phase flow equations directly (discretize-then-linearize is equivalent to linearize-then-discretize).

Taking the weighted sum of the linearized weak form equations, $\rho_n \times (\text{Eq. } 4.14) + \rho_w \times (\text{Eq. } 4.15)$, gives the discrete weak form of the "pressure" equation,

$$
\begin{aligned}
&\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla v \cdot \rho_w \rho_n \left((\bar{\lambda}_{w_S} + \bar{\lambda}_{n_S}) \mathbf{K} \nabla \bar{p}_n S'_w + (\bar{\lambda}_w + \bar{\lambda}_n) \mathbf{K} \nabla p'_n \right) \, d\Omega \\
&+ \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla v \cdot \rho_w \rho_n \left(-(\bar{\lambda}_{w_S} \bar{p}_{c_S} + \bar{\lambda}_w \bar{p}_{c_{SS}}) \mathbf{K} \nabla \bar{S}_w S'_w - \bar{\lambda}_w \bar{p}_{c_S} \mathbf{K} \nabla S'_w \right) \, d\Omega \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ \rho_w \rho_n (\bar{\lambda}_{w_S} + \bar{\lambda}_{n_S}) \mathbf{K} \nabla \bar{p}_n S'_w \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ \rho_w \rho_n (\bar{\lambda}_w + \bar{\lambda}_n) \mathbf{K} \left(\nabla p'_n + \eta_f \vec{r_p}' \right) \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ -\rho_w \rho_n (\bar{\lambda}_{w_S} \bar{p}_{c_S} + \bar{\lambda}_w \bar{p}_{c_{SS}}) \mathbf{K} \nabla \bar{S}_w S'_w \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ -\rho_w \rho_n \bar{\lambda}_w \bar{p}_{c_S} \mathbf{K} \left(\nabla S'_w + \eta_f \vec{r}'_S \right) \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f \left\{ \rho_w \rho_n (\bar{\lambda}_w + \bar{\lambda}_n) \mathbf{K}^T \nabla v \right\} \cdot [\![p'_n]\!] \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f \left\{ -\rho_w \rho_n \bar{\lambda}_w \bar{p}_{c_S} \mathbf{K}^T \nabla v \right\} \cdot [\![S'_w]\!] \, d\Gamma \\
&+ \sum_{f \in \Gamma_I} \int_f \rho_n g'_w + \rho_w g'_n \, d\Gamma = 0, \quad \forall v \in \mathcal{V}_{h,p}.
\end{aligned}
\tag{4.16}
$$

Similarly, the weighted difference of the linearized weak form equations, $\rho_n \bar{\lambda}_n \times (\text{Eq. } 4.14) - \rho_w \bar{\lambda}_w \times (\text{Eq. } 4.15)$, gives the discrete weak form of the "saturation" equation,

$$
\begin{aligned}
&\sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} v \rho_w \rho_n \phi (\bar{\lambda}_w + \bar{\lambda}_n) \frac{\partial S'_w}{\partial t} \, d\Omega \\
&+ \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla v \cdot \rho_w \rho_n \left(\bar{\lambda}_{w_S} \bar{\lambda}_n - \bar{\lambda}_w \bar{\lambda}_{n_S} \right) \mathbf{K} \nabla \bar{p}_n S'_w \, d\Omega
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla v \cdot \rho_w \rho_n \left(-(\bar{\lambda}_{w_S} \bar{\lambda}_n \bar{p}_{c_S} + \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_{SS}}) \mathbf{K} \nabla \bar{S}_w S'_w \right) \, d\Omega \\
&+ \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla v \cdot \rho_w \rho_n \left(-\bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_S} \mathbf{K} \nabla S'_w \right) \, d\Omega \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ \rho_w \rho_n \left(\bar{\lambda}_{w_S} \bar{\lambda}_n - \bar{\lambda}_w \bar{\lambda}_{n_S} \right) \mathbf{K} \nabla \bar{p}_n S'_w \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ -\rho_w \rho_n (\bar{\lambda}_{w_S} \bar{\lambda}_n \bar{p}_{c_S} + \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_{SS}}) \mathbf{K} \nabla \bar{S}_w S'_w \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f [\![v]\!] \cdot \left\{ -\rho_w \rho_n \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_S} \mathbf{K} \left(\nabla S'_w + \eta_f \vec{r}'_S \right) \right\} \, d\Gamma \\
&- \sum_{f \in \Gamma_I} \int_f \left\{ -\rho_w \rho_n \bar{\lambda}_w \bar{\lambda}_n \bar{p}_{c_S} \mathbf{K}^T \nabla v \right\} \cdot [\![S'_w]\!] \, d\Gamma \\
&+ \sum_{f \in \Gamma_I} \int_f \rho_n \bar{\lambda}_n g'_w - \rho_w \bar{\lambda}_w g'_n \, d\Gamma = 0, \quad \forall v \in \mathcal{V}_{h,p}.
\end{aligned}
\tag{4.17}
$$

4.3 Modification to discretization

The goal of this analysis is to find the appropriate stabilization terms, g_w and g_n, such that the linearized weak forms for the pressure and saturation equations given by Eq. 4.16 and Eq. 4.17 are consistent with the standard DG discretizations of the elliptic pressure equation in Eq. 4.7 and the advection-diffusion equation for saturation in Eq. 4.8 respectively.

All of the terms in Eq. 4.16, except for the last integral involving g'_w and g'_n, are consistent with a centered (BR2) DG discretization of the elliptic pressure equation in Eq. 4.7. Assuming that a centered discretization is desired for the purely elliptic pressure equation, this implies that the last integral in Eq. 4.16 should be zero, producing the condition,

$$\rho_n g'_w = -\rho_w g'_n. \tag{4.18}$$

Next, consider the DG discretization of the linearized saturation equation in

Eq. 4.9,

$$\begin{aligned}\sum_{\kappa\in\mathcal{T}_h}\int_\kappa v\frac{\partial S'_w}{\partial t}\,d\Omega - \sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla v\cdot\vec{\bar{V}}S'_w\,d\Omega + \sum_{\kappa\in\mathcal{T}_h}\int_\kappa \nabla v\cdot\bar{\mathbf{D}}\nabla S'_w\,d\Omega \\ + \sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\vec{\hat{F}}\,d\Gamma \\ - \sum_{f\in\Gamma_I}\int_f \llbracket v\rrbracket\cdot\left\{\bar{\mathbf{D}}\left(\nabla S'_w + \eta_f\vec{r}'_S\right)\right\}\,d\Gamma - \sum_{f\in\Gamma_I}\int_f \left\{\bar{\mathbf{D}}^T\nabla v\right\}\cdot\llbracket S'_w\rrbracket\,d\Gamma = 0,\end{aligned}\tag{4.19}$$

where the numerical flux $\vec{\hat{F}}(S'^+_w, S'^-_w; \vec{n}^+)$ is the exact upwind (Godunov) flux given by,

$$\vec{\hat{F}} = \left\{\vec{\bar{V}}S'_w\right\} + \frac{1}{2}\left|\vec{\bar{V}}\cdot\vec{n}^+\right|\llbracket S'_w\rrbracket\,.\tag{4.20}$$

Multiplying Eq. 4.19 by $\rho_w\rho_n\phi(\bar{\lambda}_w+\bar{\lambda}_n)$ and comparing the result with Eq. 4.17 shows that the following condition needs to be satisfied on each face $f\in\Gamma_I$ in order for the two equations to match each other,

$$\begin{aligned}\int_f \rho_w\rho_n\phi(\bar{\lambda}_w+\bar{\lambda}_n)\,\llbracket v\rrbracket\cdot\vec{\hat{F}}\,d\Gamma = &-\int_f \llbracket v\rrbracket\cdot\left\{\rho_w\rho_n\left(\bar{\lambda}_{w_S}\bar{\lambda}_n - \bar{\lambda}_w\bar{\lambda}_{n_S}\right)\mathbf{K}\nabla\bar{p}_n S'_w\right\}\,d\Gamma \\ &-\int_f \llbracket v\rrbracket\cdot\left\{-\rho_w\rho_n(\bar{\lambda}_{w_S}\bar{\lambda}_n\bar{p}_{c_S} + \bar{\lambda}_w\bar{\lambda}_n\bar{p}_{c_{SS}})\mathbf{K}\nabla\bar{S}_w S'_w\right\}\,d\Gamma \\ &+\int_f \rho_n\bar{\lambda}_n g'_w - \rho_w\bar{\lambda}_w g'_n\,d\Gamma.\end{aligned}\tag{4.21}$$

Substituting the expression for $\vec{\hat{F}}$ and canceling the average fluxes on both sides yields,

$$\int_f \llbracket v\rrbracket\cdot\frac{1}{2}\rho_w\rho_n\phi(\bar{\lambda}_w+\bar{\lambda}_n)\left|\vec{\bar{V}}\cdot\vec{n}^+\right|\llbracket S'_w\rrbracket\,d\Gamma = \int_f \rho_n\bar{\lambda}_n g'_w - \rho_w\bar{\lambda}_w g'_n\,d\Gamma.\tag{4.22}$$

Substituting the constraint in Eq. 4.18 into the right-hand side of the equation above and simplifying gives,

$$\int_f g'_w\,d\Gamma = \quad \int_f \frac{1}{2}\llbracket v\rrbracket\cdot\rho_w\phi\left|\vec{\bar{V}}\cdot\vec{n}^+\right|\llbracket S'_w\rrbracket\,d\Gamma,\tag{4.23}$$

$$\int_f g'_n \; d\Gamma = -\int_f \frac{1}{2} [\![v]\!] \cdot \rho_n \phi \left| \vec{\bar{V}} \cdot \vec{n}^+ \right| [\![S'_w]\!] \;\, d\Gamma. \tag{4.24}$$

Therefore, under the assumption of $\left[\!\left[\bar{S}_w\right]\!\right] = 0$, the nonlinear form of the stabilization terms required for upwinding can be inferred as,

$$\int_f g_w \; d\Gamma = \quad \int_f \frac{1}{2} [\![v]\!] \cdot C_w(\mathbf{u}^{\pm}_{h,p}, \nabla \mathbf{u}^{\pm}_{h,p}; \vec{n}^+) \, [\![S_w]\!] \;\, d\Gamma, \tag{4.25}$$

$$\int_f g_n \; d\Gamma = -\int_f \frac{1}{2} [\![v]\!] \cdot C_n(\mathbf{u}^{\pm}_{h,p}, \nabla \mathbf{u}^{\pm}_{h,p}; \vec{n}^+) \, [\![S_w]\!] \;\, d\Gamma, \tag{4.26}$$

where $C_\alpha(\mathbf{u}^{\pm}_{h,p}, \nabla \mathbf{u}^{\pm}_{h,p}; \vec{n}^+)$ is given by,

$$\begin{aligned} C_\alpha = \max_{\mathbf{u}_{h,p} \in \{\mathbf{u}^+_{h,p}, \mathbf{u}^-_{h,p}\}} \Bigg(\Bigg| & - \rho_\alpha \left(\frac{\lambda_{w_S} \lambda_n - \lambda_w \lambda_{n_S}}{\lambda_w + \lambda_n} \right) \mathbf{K} \left(\nabla p_n + \vec{r}_p \right) \cdot \vec{n}^+ \\ & + \rho_\alpha \left(\frac{\lambda_{w_S} \lambda_n p_{c_S} + \lambda_w \lambda_n p_{c_{SS}}}{\lambda_w + \lambda_n} \right) \mathbf{K} \left(\nabla S_w + \vec{r}_S \right) \cdot \vec{n}^+ \Bigg| \Bigg), \end{aligned} \tag{4.27}$$

for $\alpha \in \{w, n\}$. The addition of the stabilization terms g_w and g_n modifies the BR2 operator for diffusive fluxes given in Eq. 2.13 as follows,

$$\mathcal{R}^{\text{diff-upwind}}_{h,p}(\mathbf{u}; \mathbf{v}) = \mathcal{R}^{\text{diff}}_{h,p}(\mathbf{u}; \mathbf{v}) + \sum_{f \in \Gamma_I} \int_f \mathbf{g}(\mathbf{v}^{\pm}, \mathbf{u}^{\pm}, \nabla \mathbf{u}^{\pm}; \vec{n}^+) \; d\Gamma, \tag{4.28}$$

where $\mathbf{g} = [g_w, g_n]^T$. For the rest of this thesis, the DG discretization of two-phase flow equations use the upwinded diffusion operator above, instead of the standard BR2 operator. Although the upwinding terms g_w and g_n were derived under assumptions of incompressibility, they are also successfully used for slightly compressible flows later in this thesis.

4.4 Numerical results

The semi-discrete form of the DG discretization of a linear PDE can be expressed compactly using the following system of ODEs,

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{A}\mathbf{u} = \mathbf{b}, \tag{4.29}$$

where $\mathbf{u}(t)$ is the discrete solution vector and the matrices $\mathbf{M}$ and $\mathbf{A}$ are formed from the temporal and spatial parts of the DG weak form respectively. The right-hand side vector $\mathbf{b}$ usually contains forcing function and boundary condition data. Substituting a perturbed solution $\tilde{\mathbf{u}}(t) = \mathbf{u}(t) + \mathbf{u}'(t)$ into the equation above and simplifying shows that the perturbations need to satisfy the following homogeneous equation,

$$\mathbf{M}\dot{\mathbf{u}}' + \mathbf{A}\mathbf{u}' = \mathbf{0}. \tag{4.30}$$

Solving this system of ODEs shows that the evolution of the perturbations, $\mathbf{u}'(t)$, is given by,

$$\mathbf{u}'(t) = \sum_i \hat{\mathbf{u}}'_i e^{\omega_i t}, \tag{4.31}$$

where ω_i and $\hat{\mathbf{u}}'_i$ are the generalized eigenvalues and eigenvectors, respectively, of the following generalized eigenvalue problem,

$$\mathbf{A}\hat{\mathbf{u}}' = -\omega\mathbf{M}\hat{\mathbf{u}}'. \tag{4.32}$$

In this section, the stability of the DG discretization of the linearized equations (Eqs. 4.14 - 4.15) is evaluated numerically by forming the $\mathbf{M}$ and $\mathbf{A}$ matrices, and checking if the real components of the generalized eigenvalues ω are non-positive.

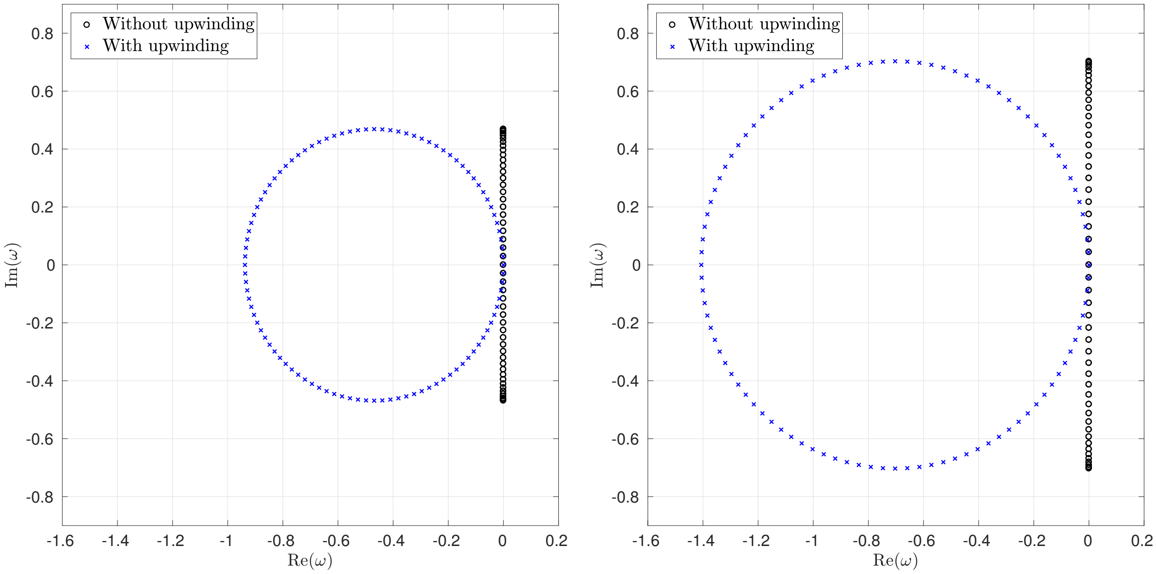
Consider a 1D incompressible two-phase flow problem in the domain $x \in [0, 100]$, with the following flow parameters,

$$\rho_w = 62.4 \text{ lb/ft}^3, \qquad \rho_n = 52.1 \text{ lb/ft}^3,$$

$$\begin{aligned} \phi &= 0.3, & \mathbf{K} &= 200\mathbf{I} \text{ mD}, \\ k_{rw}(S_w) &= S_w^2, & k_{rn}(S_w) &= (1-S_w)^2, \\ \mu_w &= 1 \text{ cP}, & \mu_n &= 2 \text{ cP}. \end{aligned}$$

Capillary effects are ignored in this problem (i.e. $p_c = 0$) in order to test for the worst-case of a purely hyperbolic saturation equation. The mean solution about which the two-phase flow equations are linearized is chosen to be a linear pressure and constant saturation solution, which satisfies the nonlinear two-phase flow equations. The DG weak form given in Eqs. 4.14 - 4.15 is then used to discretize the linearized problem with the above parameters, on a 1D grid with 100 uniformly sized elements. Figure 4-2 shows the generalized eigenvalues obtained from a piecewise constant (P0) DG discretization of the linearized two-phase flow equations with periodic boundary conditions, for two different mean solutions. The black circles, which represent the eigenvalues of the standard BR2 discretization without the upwinding modification, lie on the imaginary axis showing that the system is only neutrally stable. This is similar to the result obtained when a linear advection equation is discretized with a central difference scheme. However, with the addition of the linearized upwinding terms g'_w and g'_n given in Eqs. 4.23 - 4.24, the generalized eigenvalues move off the imaginary axis to form a circle in the stable left-half of the plane, as shown by the blue crosses. Figure 4-3 shows the results for a piecewise linear (P1) DG discretization, where the upwinding terms stabilize the system in a similar manner. Note that the figures only show the finite eigenvalues corresponding to the saturation equation, since the eigenvalues of the elliptic pressure equation do not exist.

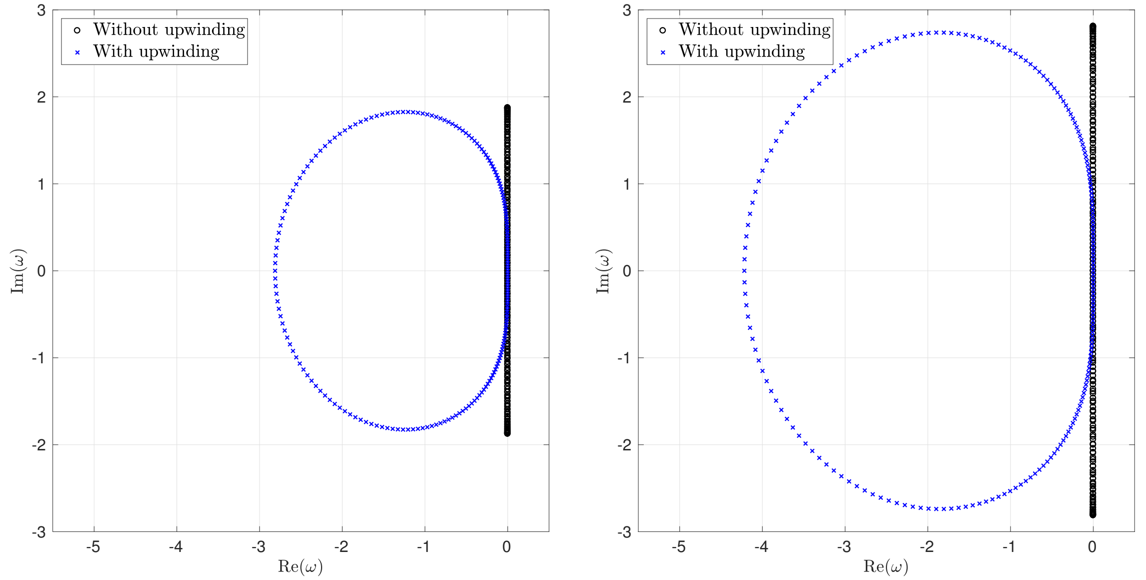
The effect of the nonlinear upwinding terms g_w and g_n (Eqs. 4.25 - 4.26) on the DG discretization of the nonlinear two-phase equations is investigated below, using the same test problem given above. Let $\mathbf{J}(\bar{\mathbf{u}}_{h,p})$ be the Jacobian matrix that arises from the DG discretization of the nonlinear two-phase equations (Eqs. 4.12 - 4.13), when evaluated about a discrete solution $\bar{\mathbf{u}}_{h,p}$. Further, let $\mathbf{J}_t(\bar{\mathbf{u}}_{h,p})$ and $\mathbf{J}_x(\bar{\mathbf{u}}_{h,p})$ be the temporal and spatial contributions to the Jacobian matrix, such that $\mathbf{J}(\bar{\mathbf{u}}_{h,p}) = \mathbf{J}_t(\bar{\mathbf{u}}_{h,p}) + \mathbf{J}_x(\bar{\mathbf{u}}_{h,p})$. Then, if the discrete solution $\bar{\mathbf{u}}_{h,p}$ is equal to the mean



(a) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.2$

(b) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.5$

Figure 4-2: Generalized eigenvalues of a DG P0 discretization of the linearized two-phase equations with periodic BCs, for different mean solutions



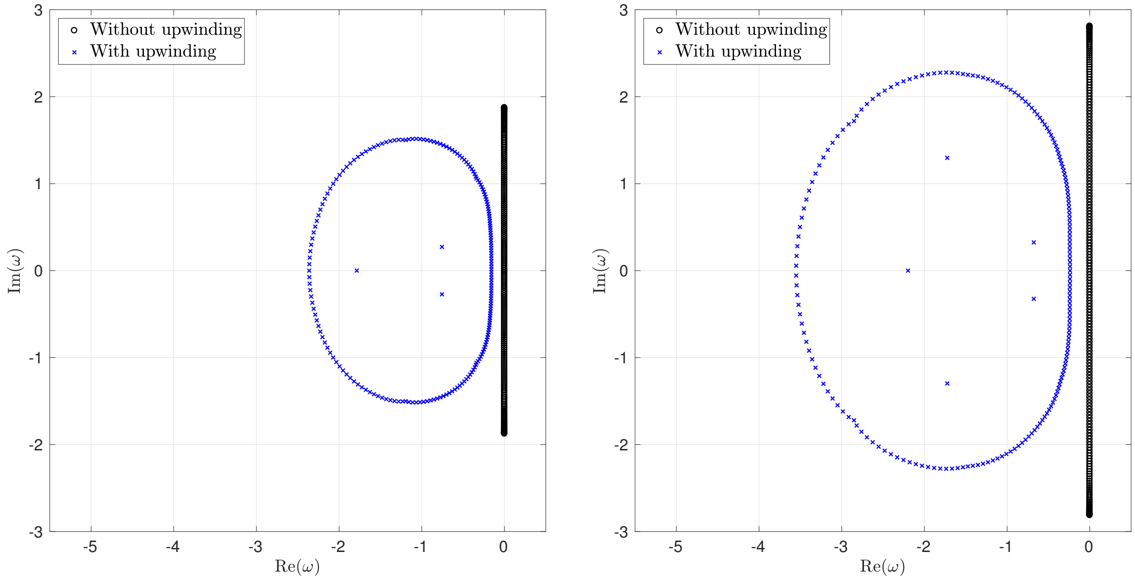
(a) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.2$

(b) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.5$

Figure 4-3: Generalized eigenvalues of a DG P1 discretization of the linearized two-phase equations with periodic BCs, for different mean solutions

solution about which the continuous equations are linearized (i.e. $\bar{\mathbf{u}}_{h,p} = [\bar{p}_n, \bar{S}_w]$), it is expected that the generalized eigenvalues of the problem $[\mathbf{J}_x\hat{\mathbf{u}}' = -\omega\mathbf{J}_t\hat{\mathbf{u}}']$ will be equal to those obtained for the linearized two-phase flow equations.

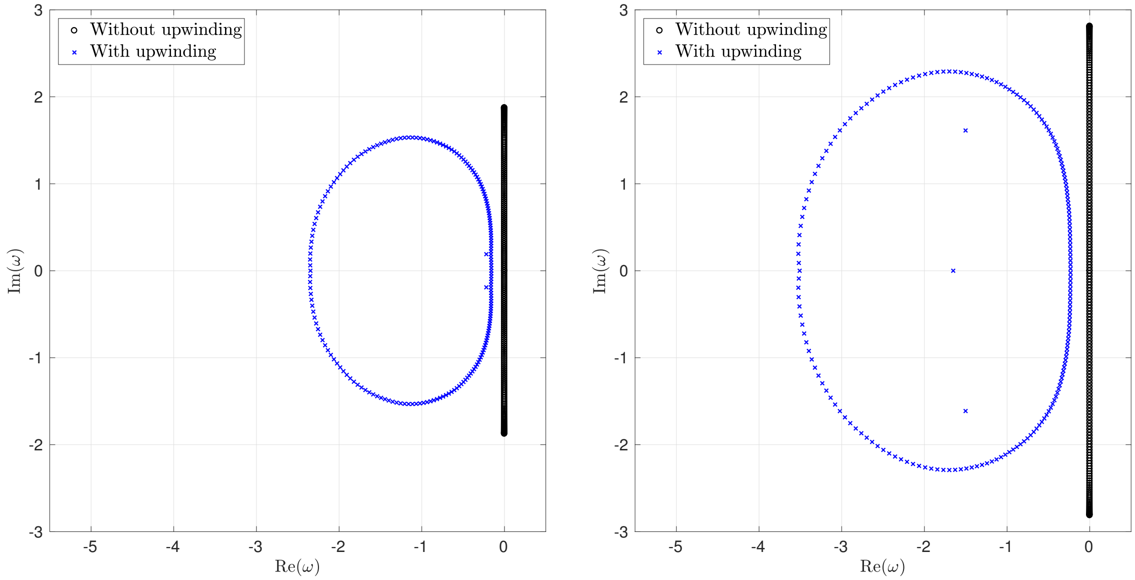
In order to test this, the $\mathbf{J}_t$ and $\mathbf{J}_x$ matrices from a P1 DG discretization of the nonlinear equations are evaluated at a discrete solution that is an L^2-projection of the mean solution onto $\mathcal{V}_{h,p}$. A P1 DG discretization is used since a linear pressure and constant saturation solution can be represented exactly in the P1 discrete space. Figure 4-4 shows the generalized eigenvalues obtained from a P1 discretization of the linearized two-phase equations with Dirichlet BCs for p_n' at both boundaries, and a Dirichlet BC for S_w' at the left boundary. The addition of Dirichlet BCs shifts the eigenvalues of the upwinded cases away from the origin. Figure 4-5 shows the generalized eigenvalues obtained from a P1 discretization of the nonlinear two-phase flow equations, where the Jacobian matrices are evaluated at the same mean solution. The similarity of the eigenvalues in Figures 4-4 and 4-5 confirms that the discretization of the nonlinear upwinding terms g_w and g_n behave as expected when linearized.



(a) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.2$

(b) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.5$

Figure 4-4: Generalized eigenvalues of a DG P1 discretization of the linearized two-phase equations with Dirichlet BCs, for different mean solutions



(a) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.2$

(b) $\frac{d\bar{p}_n}{dx} = -0.25, \bar{S}_w = 0.5$

Figure 4-5: Generalized eigenvalues of a DG P1 discretization of the nonlinear two-phase equations with Dirichlet BCs, evaluated at a P1 L^2-projection of the mean solution

Chapter 5

A distributed bottom-hole pressure well model

This chapter presents a new bottom-hole pressure well model for reservoir simulation. In contrast to Peaceman-type well models, the proposed well model relates the volumetric flow rate and the bottom-hole pressure of the well to the reservoir pressure through a distributed source term. The discretization-independent formulation of this well model makes it readily applicable to finite element discretizations on arbitrarily unstructured meshes, and hence serves as a key component of the adaptive space-time DG framework developed in this thesis.

5.1 Review of Peaceman's well model

The derivation of a mathematical well model is based on the assumption that the flow is radial in the neighborhood of a well. In simplified scenarios, the flow can then be described using analytic equations. In particular, the rest of this section assumes a single-phase, incompressible, steady-state flow in a homogeneous, isotropic reservoir. Peaceman's model [113] represents the well as a single line source which, without loss of generality, is located at the origin and aligned with the z-axis. The mass conservation equation for this flow in cylindrical coordinates is given by,

$$\nabla \cdot (\rho \vec{u}(r)) = \rho Q_{\text{line}} \delta(r), \tag{5.1}$$

where ρ is the fluid density, $\vec{u}(r)$ is the fluid velocity, $Q_{\text{line}} = Q/L_z$ is the strength of the line source, and $\delta(r)$ is the Dirac delta function. Q represents the volumetric flow rate of the well, and L_z is the constant depth of the reservoir. Darcy's law for a homogeneous, isotropic reservoir without the gravity terms is given by,

$$\vec{u}(r) = -\frac{1}{\mu}\mathbf{K}\nabla p = -\frac{K}{\mu}\frac{dp}{dr}\vec{e_r}, \tag{5.2}$$

where $\mathbf{K} = K\mathbf{I}$ is the isotropic absolute permeability tensor, μ is the constant fluid viscosity, $p(r)$ is the fluid pressure, and $\vec{e_r}$ is the unit radial vector. Integrating Eq. 5.1 over a small cylindrical volume enclosing the origin and applying the divergence theorem yields,

$$\begin{gathered}\int_0^{L_z}\int_0^{2\pi} \rho \vec{u} \cdot \vec{e_r}\ r\ d\theta\ dz = \int_0^{L_z}\int_0^{2\pi}\int_0^{r} \rho Q_{\text{line}} \delta(r)\ r\ dr\ d\theta\ dz, \\ \vec{u} \cdot \vec{e_r} = \frac{Q}{2\pi L_z r}. \end{gathered} \tag{5.3}$$

Substituting the Darcy velocity from Eq. 5.2 into the above equation gives,

$$\frac{dp}{dr} = -\frac{\mu}{2\pi K L_z}\frac{Q}{r}, \tag{5.4}$$

which can then be integrated radially from the well-bore radius r_w to some radial distance r, to produce the following analytic expression for the fluid pressure at a radial distance r,

$$p_{\text{analytic}}(r) = p(r_w) - \frac{\mu Q}{2\pi K L_z} \ln\left(\frac{r}{r_w}\right). \tag{5.5}$$

Peaceman then uses the above expression to obtain a relationship between the flowing bottom-hole pressure of the well and the pressure calculated for the cell that contains the well using a finite volume discretization. The application of a two-point flux approximation (TPFA) finite volume scheme to the single-phase pressure equation on a structured 2D grid with uniform grid spacing h yields,

$$\frac{\rho K L_z}{\mu}\left(4p_{i,j} - p_{i-1,j} - p_{i+1,j} - p_{i,j-1} - p_{i,j+1}\right) = \rho Q, \tag{5.6}$$

where $p_{i,j}$ is the pressure value in the cell containing the well. Due to the symmetry of the pressure field around the well cell (i.e. $p_{i-1,j} = p_{i+1,j} = p_{i,j-1} = p_{i,j+1}$), the above equation simplifies to,

$$\frac{4KL_z}{\mu}\left(p_{i,j} - p_{i+1,j}\right) = Q. \tag{5.7}$$

If it is assumed that the pressure in the adjacent cells is computed accurately according to Eq. 5.5, it follows that,

$$p_{i+1,j} = p_{\text{analytic}}(h) = p_{wf} - \frac{\mu Q}{2\pi K L_z}\ln\left(\frac{h}{r_w}\right), \tag{5.8}$$

where $p_{wf} = p(r_w)$ is referred to as the bottom-hole pressure. Substituting the above equation in Eq. 5.7 gives,

$$\begin{aligned} p_{i,j} &= p_{wf} - \frac{\mu Q}{2\pi K L_z}\ln\left(\frac{h}{r_w}\right) + \frac{\mu Q}{4KL_z}, \\ &= p_{wf} - \frac{\mu Q}{2\pi K L_z}\left(\ln\left(\frac{h}{r_w}\right) - \frac{\pi}{2}\right), \\ &= p_{wf} - \frac{\mu Q}{2\pi K L_z}\ln\left(\frac{r_e}{r_w}\right), \end{aligned} \tag{5.9}$$

where r_e is the equivalent well radius defined as,

$$r_e = e^{-\pi/2}h \approx 0.20788h. \tag{5.10}$$

Comparing Eq. 5.5 with the right-hand side of Eq. 5.9 shows that the equivalent well radius is the radius at which the steady state flowing pressure for the actual well, $p_{\text{analytic}}(r_e)$, is equal to the numerically computed pressure for the well cell, $p_{i,j}$. If the bottom-hole pressure is known, Eq. 5.9 can be inverted to produce Peaceman's well model for the unknown volumetric flow rate as follows,

$$Q = -\frac{2\pi K L_z}{\mu \ln\left(\frac{r_e}{r_w}\right)} \left(p_{i,j} - p_{wf}\right). \tag{5.11}$$

It's worth noting that although the definition of r_e given in Eq. 5.10 is widely used in many textbooks, papers and numerical simulators, it is still an approximation since Eq. 5.8 is largely an assumption. In [114], Peaceman provides the following alternate expression for the equivalent well radius r_e, which is derived by analyzing the pressure distribution for a TPFA finite volume method on an infinite grid using Fourier analysis,

$$r_e = \frac{e^{-\gamma}}{2\sqrt{2}} h \approx 0.19851h, \tag{5.12}$$

where $\gamma = 0.577215...$ is the Euler-Mascheroni constant. In addition to the r_e definitions given by Eqs. 5.10 and 5.12, the rule of thumb $r_e = 0.2h$ is also prevalent in the literature.

The effect of these slightly different r_e definitions on the accuracy of Peaceman's well model is investigated below. The steady single-phase pressure equation used in Peaceman's analysis is discretized using a finite volume scheme with a two-point flux approximation on a square grid with uniform spacing. The isotropic permeability is specified to be $K = 200$ mD and the fluid viscosity $\mu = 1$ cP. A single production well is located at the origin of a square domain $\Omega_s = [-L, L] \times [-L, L]$, where $L = 1000$ ft. The bottom-hole pressure and the well-bore radius of the well is set to $p_{wf} = 2000$ psi and $r_w = 2$ inches respectively. Dirichlet boundary conditions are imposed on all four boundaries of the domain, where the pressure is set according to the analytic solution $p_{\text{analytic}}(r)$ given by Eq. 5.5, for a specified "true" flow rate $Q = -1000$ ft^3/day. The well is then modeled using Peaceman's well model with the three different r_e definitions, and the accuracy of each case is evaluated based on how well the predicted flow rate Q_h converges to the true flow rate Q with grid refinement. The results of this numerical investigation are given in Figure 5-1, which shows how the relative error in the predicted flow rate behaves with the average mesh size h, for different variations of Peaceman's well model.

It is clearly visible from Figure 5-1 that the versions of Peaceman's well model with $r_e = 0.2h$ (solid blue line) and $r_e = e^{-\pi/2}h$ (solid red line) are not consistent with the analytic pressure profile given in Eq. 5.5, since their flowrate errors do not decrease with uniform mesh refinement. However, the Peaceman well model using $r_e = \frac{e^{-\gamma}}{2\sqrt{2}}h$ (solid black line) yields a second-order convergence to the true value. These results present clear evidence to the fact that the definition of the equivalent well radius r_e has a profound impact on the error convergence behavior of Peaceman's well model, and that the r_e definition in Eq. 5.12 should be used instead of the other approximations when a TPFA finite volume method is applied.

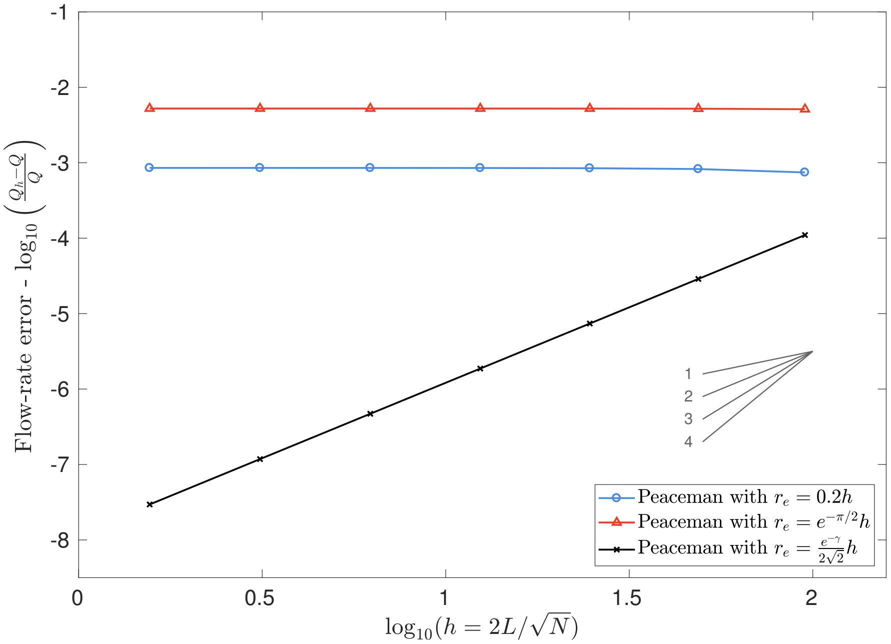


Figure 5-1: Flow rate error convergence for Peaceman's well model with different r_e definitions

5.1.1 Extensions to anisotropic media and rectangular meshes

Peaceman's well model has also been extended to the case of diagonally anisotropic permeability tensors and rectangular meshes [114]. The equivalent form of Eq. 5.5 for this general case is given by,

$$p_{\text{analytic}}(x, y) = p_{wf} - \frac{\mu Q}{2\pi\sqrt{k_{xx}k_{yy}}L_z}\ln\left(\frac{\bar{r}}{\bar{r}_w}\right), \tag{5.13}$$

where,

$$\bar{r} = \sqrt{\left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{2}} x^2 + \left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{2}} y^2}, \tag{5.14}$$

and,

$$\bar{r}_w = \frac{1}{2} r_w \left(\left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}} + \left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}}\right), \tag{5.15}$$

where k_{xx} and k_{yy} are the horizontal components of the diagonal permeability tensor $\mathbf{K} = \text{diag}(k_{xx}, k_{yy}, k_{zz})$. The $\sqrt{k_{xx}k_{yy}}$ factor in Eq. 5.13 is a result of the coordinate transformation: $x' = (k_{yy}/k_{xx})^{\frac{1}{4}}x$ and $y' = (k_{xx}/k_{yy})^{\frac{1}{4}}y$. Similarly, the general form of Eq. 5.9 is given by,

$$p_{i,j} = p_{wf} - \frac{\mu Q}{2\pi\sqrt{k_{xx}k_{yy}}L_z}\ln\left(\frac{r_e}{r_w}\right), \tag{5.16}$$

where the definition of the equivalent radius r_e is generalized to,

$$r_e = \frac{e^{-\gamma}}{2}\frac{\left((k_{yy}/k_{xx})^{1/2} h_x^2 + (k_{xx}/k_{yy})^{1/2} h_y^2\right)^{1/2}}{(k_{yy}/k_{xx})^{1/4} + (k_{xx}/k_{yy})^{1/4}}, \tag{5.17}$$

where h_x and h_y are the grid spacings in the x and y directions respectively. The reader is referred to [114] for a more detailed derivation. All subsequent numerical results obtained using Peaceman's well model in this chapter use the general model given above in Eqs. 5.16 and 5.17.

Although Peaceman's well model was originally derived for finite difference or TPFA finite volume discretizations on *uniform* rectangular meshes [114], it is being widely applied by the reservoir simulation community on increasingly complex configurations and meshes. Hence, it is desirable to study how deviations from a uniform mesh affects the accuracy of Peaceman's well model. Figure 5-3 contains the results of a grid convergence study performed on the steady, isotropic single-phase flow problem described earlier, using Peaceman's well model (Eqs. 5.16 and 5.17) with the finite volume method on rectangular meshes. The solid black line represents results obtained on uniform square meshes, whereas the solid blue, red and green lines represent results on slightly non-uniform meshes where the node locations are transformed using the exponential mapping given by,

$$x(u;\beta) = \operatorname{sgn}(u)\left(\frac{e^{\beta|u|}-1}{e^{\beta}-1}\right)L, \qquad \forall u \in [-1,1] \tag{5.18}$$

$$y(v;\beta) = \operatorname{sgn}(v)\left(\frac{e^{\beta|v|}-1}{e^{\beta}-1}\right)L, \qquad \forall v \in [-1,1] \tag{5.19}$$

where β is a parameter that characterizes the nonlinearity of the transformation. In the limit as $\beta \to 0$, Eqs. 5.18 and 5.19 reduce to the linear transformations $x = uL$ and $y = vL$ respectively. The non-uniform meshes are produced by first generating uniform meshes in the $u - v$ space, and then transforming the node coordinates to the $x - y$ space via the mappings given above, for a given value of β. This nonlinear transformation produces elements that grow in size away from the origin, as seen by the 1D grid spacings given in Figure 5-2. The flow rate predictions on these non-uniform meshes clearly show that Peaceman's well model is unable to achieve second-order convergence to the true value even with a very small deviation away (i.e. $\beta = 0.01$) from a uniformly spaced mesh. Larger β values produce larger errors in the predicted flow rate, thereby highlighting the adverse sensitivity of the well model to the non-uniformity of the mesh.

Furthermore, it can be observed from Eq. 5.11 that Peaceman's well model breaks down when $r_e \leq r_w$. For the case of a square mesh with an isotropic reservoir, this means that the local grid size h should be larger than about five times the well-bore

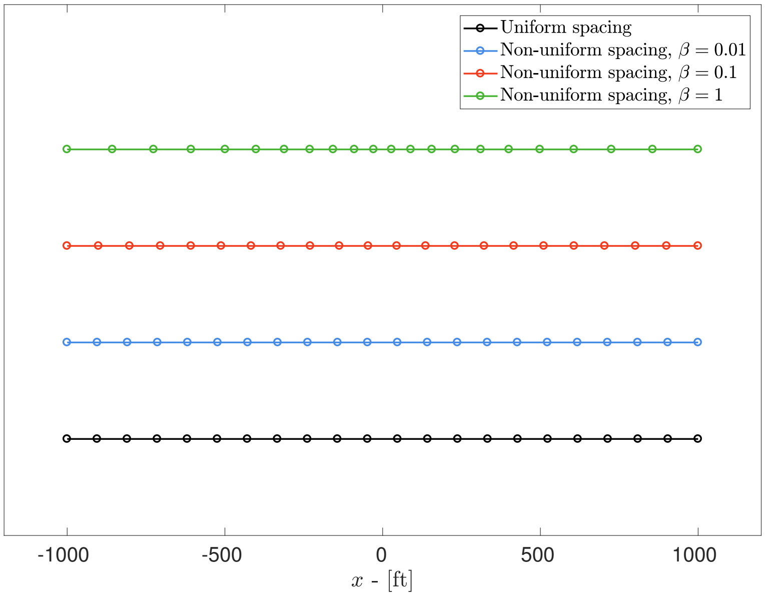


Figure 5-2: Non-uniform grid spacings in 1D for $\beta = 0.01, 0.1$ and 1

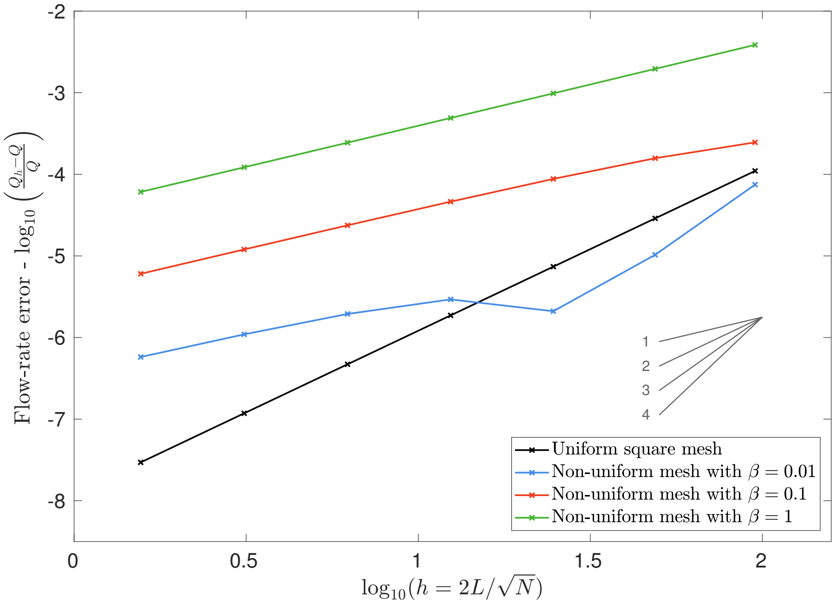


Figure 5-3: Flow rate error convergence for Peaceman's well model with FV on uniform and non-uniform meshes

radius r_w. Special care is required to ensure that this constraint is not violated on hand designed grids, and especially when this well model is used with adaptive mesh refinement algorithms, where there is lesser control over the size of individual grid cells.

5.2 Distributed well model

This section presents the derivation of an analytic expression for a source term that models the behavior of a well which, unlike Peaceman's model or its extensions, does not inherently depend on the discretization or the mesh that is used to numerically solve the pressure equation. As before, the analysis assumes a single-phase, incompressible, steady-state flow in a homogeneous, isotropic reservoir. Without loss of generality, the well is considered to be centered at the origin and aligned with the z-axis. The mass conservation equation is given by,

$$\nabla \cdot (\rho \vec{u}(r)) = \rho q(r), \tag{5.20}$$

where ρ is the fluid density, $\vec{u}(r)$ is the fluid velocity, and $q(r)$ is some source function that models the volumetric flow rate per unit volume of the well. In contrast to Peaceman's well model, where $q(r)$ is assumed to be a Dirac delta function, this analysis allows the source term to take a much more general form. The goal of this analysis is to find a particular form of the source term $q(r)$ that possesses certain desired characteristics and features.

5.2.1 Desired characteristics of a well model

This subsection provides a brief discussion of some characteristics that are deemed desirable in a well model.

- **Discretization agnostic:**
  The well model should be defined independently of the numerical method or grid that is used to solve the problem. The source term $q(r)$ should not have an explicit dependence on the grid length scales, so as to avoid any model breakdown behaviors as in the case of Peaceman's well model. Local grid length scales may also be ambiguous for certain types of grid elements or unstructured meshes, and are therefore best avoided.

- **Agreement with Peaceman:**

  The pressure profiles and flow rates produced by a new well model should agree with the results produced by Peaceman's well model for the same flow conditions. Although the pressure behavior inside the active region of the well may differ with Peaceman, a pressure probe located outside the modeled region of the well should not be able to distinguish between Peaceman's well model and the new well model. The widespread use of Peaceman's model and its validation against decades of field data has made it a valuable benchmark for any new well models.

- **Well-behaved:**

  The source term $q(r)$ should be well defined, and readily discretizable as a volumetric source term in the governing PDE. Dirac delta functions and other singular behaviors must be avoided. Furthermore, if the numerical method uses an adaptive mesh refinement scheme, any large gradients in pressure near the well center are likely to be detected by the algorithm, and would thereby cause it to increase the mesh resolution inside and near the well to resolve those pressure gradients. Therefore, it is desirable to have a model that is capable of producing an analytic pressure profile that is smooth and benign inside the modeled (active) region of the well, which would avoid such mesh refinement behaviors.

- **User controllable modeled region:**

  The user needs to be able to specify the size of the region modeled by the well model. This size is typically governed by the extent to which the user desires to "resolve" the near-well region through mesh refinement. Ideally, the size of the modeled region of the well model should be independent of any grid length scales, so that the user may decide the extent of modeled region based on geological or other physical data, and not based on the grid.

- **Smoothness:**

  When developing a well model that can also be used with high-order discretizations and mesh adaptation schemes, it is worth noting that the mesh refinement behavior for high-order methods may be driven not only by discontinuities or abrupt changes in solution value and gradients, but also in its higher derivatives. Therefore, ideally, it is desirable to have a well model that can produce infinitely smooth analytic pressure profiles in order to avoid unnecessary mesh adaptation behaviors. However, if such a well model is unrealizable, it is still preferable to be able to produce analytic pressure profiles of any specified finite level of smoothness (i.e. $\mathcal{C}^1, \mathcal{C}^2, \mathcal{C}^3, ...$).

5.2.2 Analytic equations

This analysis assumes a homogeneous, isotropic reservoir for which Darcy's law is given by Eq. 5.2. Substituting Darcy's law into Eq. 5.20 yields the following relationship between the pressure and the source term,

$$q(r) = -\frac{K}{\mu}\frac{1}{r}\frac{d}{dr}\left(r\frac{dp}{dr}\right). \tag{5.21}$$

Integrating the above equation yields the pressure distribution,

$$p(r) = p(0) - \frac{\mu}{2\pi K L_z}\int_0^r \frac{Q(r)}{r}dr. \tag{5.22}$$

where $Q(r)$ is the volumetric flow rate inside the region of radius r, defined as,

$$Q(r) \equiv 2\pi L_z \int_0^r q(r)\ r\ dr. \tag{5.23}$$

The active region of the well is set by a user controllable radius R, which can be much larger than the well-bore radius r_w. In order to produce a smooth and benign pressure profile inside the active region of the well, as discussed in Section 5.2.1, the logarithmic pressure profile present outside the well must ideally be smoothly driven to a finite value at the origin ($r = 0$). Hence, the pressure profile inside the active region may be assumed to be of the following form,

$$p^*(r) = p_{wf} + \alpha \left(1 + f_0 - f(s)\right), \qquad s = \frac{r}{R} \leq 1, \tag{5.24}$$

where p_{wf} is the bottom-hole pressure, and α is a free parameter. The "activation" function $f(s)$ satisfies $f(0) = 0$ and $f(1) = 1$, such that the pressure at radius $r = R$ is equal to $p_{wf} + \alpha f_0$ by design, where f_0 is a constant parameter that accounts for the fact that the well is active over a larger radius than r_w. If $R = r_w$, then $f_0 = 0$. The exact form of $f(s)$ and the value of f_0 is to be determined later. The corresponding form of the source term is found by taking derivatives of the assumed pressure profile $p^*(r)$ and substituting them into Eq. 5.21, which yields,

$$\begin{aligned} q^*(r) &= -\frac{K}{\mu}\alpha \left(-\frac{1}{r}\frac{f'(s)}{R} - \frac{f''(s)}{R^2}\right) \\ &= \frac{K}{\mu}\frac{\alpha}{R^2}\left(\frac{1}{s}f'(s) + f''(s)\right), \qquad r \leq R. \end{aligned} \tag{5.25}$$

The free parameter α can be eliminated by using the pressure profile assumed in Eq. 5.24, yielding the following source term function,

$$\boxed{q(r) = \begin{cases} \frac{K}{\mu}\left(\frac{p(r)-p_{wf}}{1+f_0-f(s)}\right)\frac{1}{R^2}\left(\frac{1}{s}f'(s) + f''(s)\right), & r \leq R, \\ 0, & r > R, \end{cases}} \tag{5.26}$$

which can be readily implemented as a solution-dependent source term in any discrete numerical method. Substituting the assumed pressure profile in to Eq. 5.26 and integrating gives the volumetric flow rate inside the well,

$$\begin{aligned} Q(r) &= 2\pi L_z \int_0^r q(r) r dr, \\ &= 2\pi L_z \frac{K}{\mu}\alpha s f'(s), \qquad r \leq R, \end{aligned} \tag{5.27}$$

under the assumption that $f'(0)$ is finite. Note that the volumetric flow rate is constant beyond the radius R since the source term $q(r)$ is zero outside the well.

Hence, the volumetric flow rate in the complete domain is given by,

$$Q(r) = \begin{cases} 2\pi L_z \frac{K}{\mu} \alpha s f'(s), & r \leq R, \\ \\ 2\pi L_z \frac{K}{\mu} \alpha f'(1), & r > R. \end{cases} \tag{5.28}$$

The volumetric flow rate function above can be substituted into Eq. 5.22 to solve for the pressure distribution in a piecewise manner as shown below.

If $r \leq R$,

$$p(r) = p_{wf} + \alpha(1 + f_0 - f(s)). \tag{5.29}$$

If $r > R$,

$$p(r) = p_{wf} + \alpha f_0 - \alpha f'(1) \ln(s). \tag{5.30}$$

The absence of a source term outside the radius R causes the pressure to take a logarithmic form in that region. The value of f_0 is derived by imposing the condition that the pressure far away from the well needs to remain constant for any choice of model radius R. First, consider the case where $R = r_w$, for which Eq. 5.30 simplifies to,

$$p(r) = p_{wf} - \alpha f'(1) \ln\left(\frac{r}{r_w}\right). \tag{5.31}$$

Next, consider the case where $R > r_w$, for which the pressure is given by,

$$p(r) = p_{wf} + \alpha f_0 - \alpha f'(1) \ln\left(\frac{r}{R}\right). \tag{5.32}$$

Imposing the equality of the two cases above for large r yields the following expression for f_0,

$$f_0 = -f'(1) \ln\left(\frac{R}{r_w}\right). \tag{5.33}$$

Up to this point in the analysis, the exact form of $f(s)$ was left undefined. One possible choice for $f(s)$ is to assume a polynomial form that imposes additional smoothness constraints on the resulting pressure profile. In particular, it is desirable to have a continuity of pressure derivatives across the $r = R$ boundary, such that the pressure profile inside the well-bore transitions to the logarithmic profile outside in a smooth, continuous manner. Furthermore, vanishing higher derivatives of pressure at the origin allows for greater continuity of the solution across the well center.

The pressure profiles given by Eqs. 5.29 and 5.30 show that the pressure value is continuous across $r = R$, provided that $f(1) = 1$. The k-th radial derivative of pressure (for $k > 0$) is given by,

$$\frac{d^k p}{dr^k} = -\frac{\alpha}{R^k}\frac{d^k f}{ds^k}, \qquad r \leq R, \tag{5.34}$$

$$\frac{d^k p}{dr^k} = -\frac{\alpha}{R^k} f'(1)\left(\frac{(-1)^{k+1}(k-1)!}{s^k}\right), \qquad r > R. \tag{5.35}$$

The limiting values of the radial pressure gradient as evaluated from inside and outside the well-bore radius are obtained by evaluating the above equations for $k = 1$ at $r = R$, which yields,

$$\left.\frac{dp}{dr}\right|_{r=R^-} = \left.\frac{dp}{dr}\right|_{r=R^+} = -\frac{\alpha}{R}f'(1). \tag{5.36}$$

The equality of the gradients above show that the radial pressure gradient is continuous across the $r = R$ boundary, regardless of the form of $f(s)$. However, imposing the continuity of second or higher order pressure derivatives across $r = R$ requires $f(s)$ to satisfy the additional constraints given below,

$$\left.\frac{d^k f}{ds^k}\right|_{s=1} = f'(1)\ (-1)^{k+1}(k-1)!, \qquad \text{for } k \geq 2. \tag{5.37}$$

The continuity of the pressure profile and its derivatives across the well center is a desirable property for mesh adaptation schemes, since it avoids unnecessary adaptation near the well center. This requires $p(r)$, and consequently $f(s)$, to have vanishing odd derivatives at $r = 0$. Further imposing zero even derivatives of $f(s)$ at $s = 0$ produces "flatter" and more benign pressure profiles inside the well. Based on these observations, requiring all derivatives of $f(s)$ to vanish at the origin seems an attractive option. However, if both $f'(s)$ and $f''(s)$ tend to zero as $s \to 0$, then it is evident from Eq. 5.26 that the source term $q(r)$ also tends to zero as $s \to 0$, giving rise to an undesirable stagnant flow near the center of the well. In multi-phase flows for example, this could cause a particular fluid phase to remain trapped near the center of the well, producing unrealistic solutions. Fortunately, this situation can be easily avoided by allowing only $f''(s)$ to be nonzero at $s = 0$. The constraints on $f(s)$ at the origin are therefore given by,

$$\left.\frac{d^k f}{ds^k}\right|_{s=0} = 0, \qquad \text{for } k \geq 1 \text{ and } k \neq 2. \tag{5.38}$$

The value constraints $f(0) = 0$ and $f(1) = 1$, and the derivative constraints in Eqs. 5.37 and 5.38 show that a $\mathcal{C}^m$-continuous pressure profile requires a total of $2m$ constraints on $f(s)$, for $m > 1$. Therefore, one possible choice for the function $f(s)$ is to consider polynomial functions of the form,

$$f_m(s) = \begin{cases} \sum_{k=0}^{2} a_k s^k, & \text{for } m = 1, \\ \\ \sum_{k=0}^{2m-1} a_k s^k, & \text{for } m > 1, \end{cases} \tag{5.39}$$

where the constants $a_0, a_1, ...a_{2m-1}$ are tuned to satisfy the value and derivative constraints. Table 5.1 gives a list of polynomial functions $f_m(s)$ derived in this manner for pressure solutions that are continuous up to the sixth derivative ($m = 6$). Figure 5-4 contains a plot of the activation functions given in Table 5.1.

Figures 5-5 to 5-8 provide comparisons of the pressure and flow rate distributions obtained by Peaceman's well model and the proposed distributed well model. All the profiles shown in the figures are analytically derived for a single well located at the origin in a reservoir of unit depth, with constant, isotropic permeability $K = 200$

Order of pressure continuity (m)	Polynomial activation function
$\mathcal{C}^1$	$f_1(s) = s^2$
$\mathcal{C}^2$	$f_2(s) = \frac{9}{5}s^2 - \frac{4}{5}s^3$
$\mathcal{C}^3$	$f_3(s) = \frac{100}{57}s^2 - \frac{25}{19}s^4 + \frac{32}{57}s^5$
$\mathcal{C}^4$	$f_4(s) = \frac{735}{424}s^2 - \frac{147}{53}s^5 + \frac{1225}{424}s^6 - \frac{45}{53}s^7$
$\mathcal{C}^5$	$f_5(s) = \frac{4536}{2635}s^2 - \frac{3528}{527}s^6 + \frac{31104}{2635}s^7 - \frac{3969}{527}s^8 + \frac{896}{527}s^9$
$\mathcal{C}^6$	$f_6(s) = \frac{50820}{29657}s^2 - \frac{522720}{29657}s^7 + \frac{1334025}{29657}s^8 - \frac{1355200}{29657}s^9 + \frac{640332}{29657}s^{10} - \frac{117600}{29657}s^{11}$

Table 5.1: Polynomial activation functions $f_m(s)$ for different orders of pressure continuity

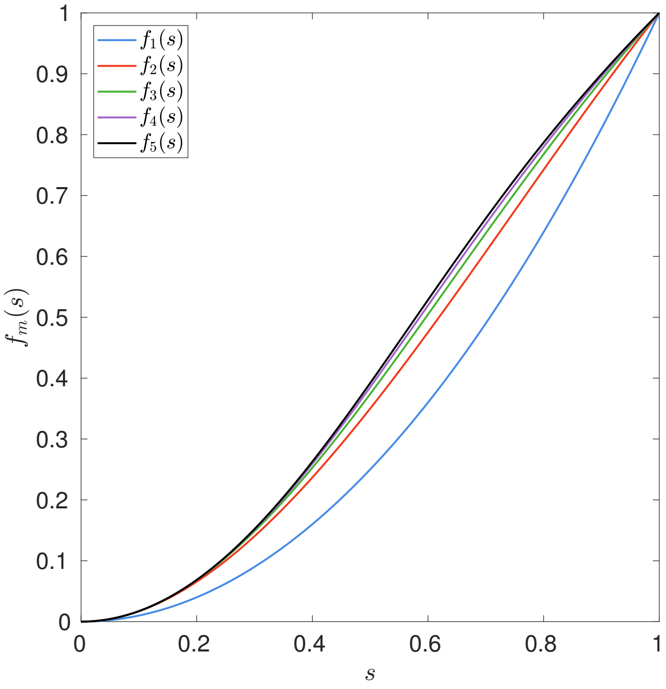
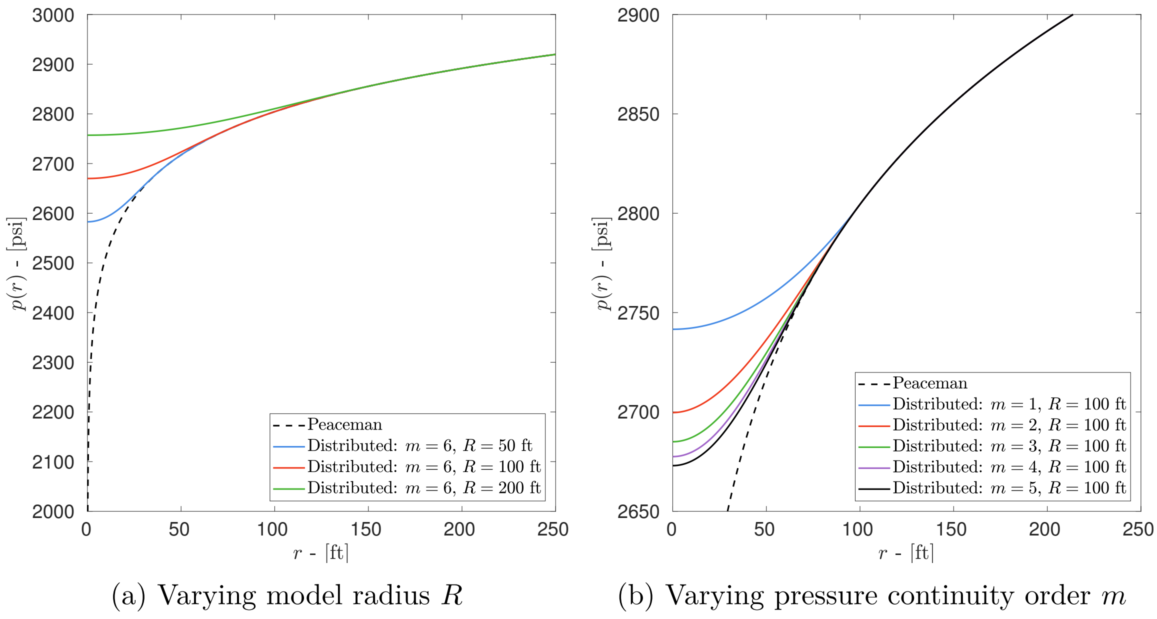


Figure 5-4: Polynomial activation functions $f_m(s)$ for different orders of pressure continuity

mD and fluid viscosity $\mu = 1$ cP. Both well models are set up to have a bottom-hole pressure $p_{wf} = 2000$ psi at a well-bore radius of $r_w = 2$ inches, and a total outflow rate of 1000 ft^3/day.

Figure 5-5a compares Peaceman's pressure profile obtained using Eq. 5.5 (black dashed line) with the profiles obtained from Eqs. 5.29 - 5.30 for a distributed well



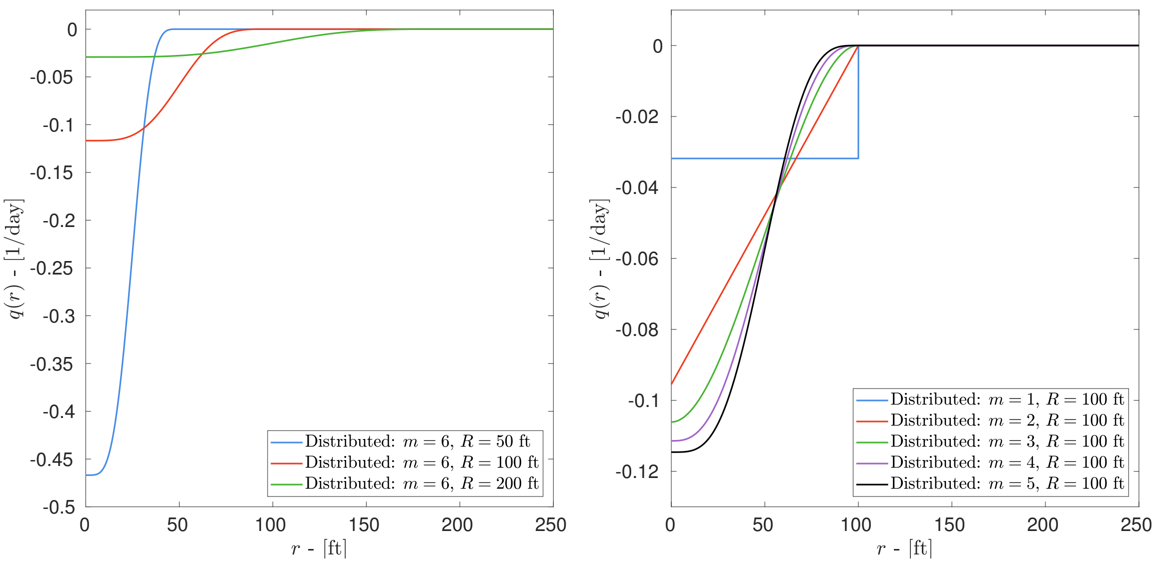
(a) Varying model radius R

(b) Varying pressure continuity order m

Figure 5-5: Comparison of analytic pressure profiles between Peaceman and the distributed model

with $\mathcal{C}^6$ pressure continuity ($m = 6$) at different model radii (solid lines). It is clearly visible that the distributed wells produce pressure profiles that have the form of $f_6(s)$ inside the radius R, which smoothly transition to the logarithmic Peaceman pressure profile outside the radius R. Figure 5-5b shows the pressure profiles obtained from distributed wells of the same model radius R, but with different orders of pressure continuity (varying m). All the solid lines match up with Peaceman's profile for $r \geq R = 100$ ft, but the differences in the continuity of pressure derivatives cause them to be different inside the active region of the well.

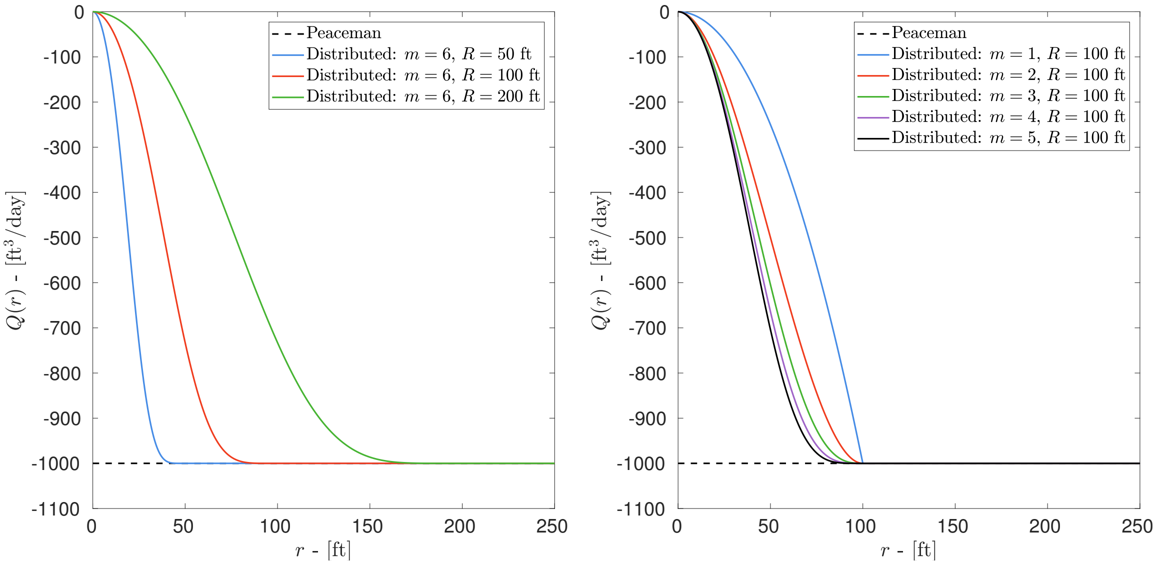
Figures 5-6a and 5-6b show the source term distributions for distributed well models of different model radii and pressure continuity orders, respectively. The source term of Peaceman's well model is a Dirac delta function at the origin, and hence is not shown on these figures. The distributed well sources are non-zero (active) only within the specified radius R, and models with larger m exhibit smoother transitions to zero at $r = R$. Similarly, Figures 5-7a and 5-7b contain plots of the volumetric flow rate distributions of each well model plotted against the radial distance away from the center of the well. Recall that the volumetric flow rate $Q(r)$ is the radial integral



(a) Varying model radius R

(b) Varying pressure continuity order m

Figure 5-6: Comparison of analytic source term distributions



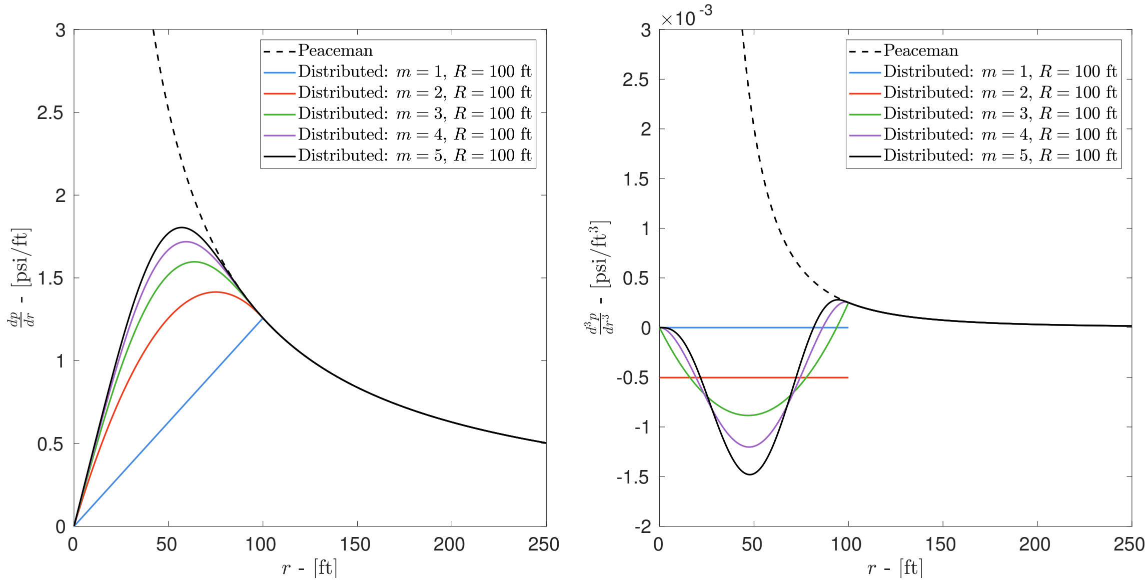
(a) Varying model radius R

(b) Varying pressure continuity order m

Figure 5-7: Comparison of analytic volumetric flowrate distributions

of the source term $q(r)$, as given in Eq 5.23. Therefore, Peaceman's model (black dashed line) has a constant flowrate equal to the specified value of -1000 ft^3/day, whereas the distributed wells (solid lines) have monotonously decreasing profiles that reach the specified flow rate at $r = R$ and remain constant for $r > R$.

Figure 5-8a shows that the first derivative of pressure is continuous across $r =$



(a) First pressure derivative

(b) Third pressure derivative

Figure 5-8: Comparison of analytic pressure derivatives for different orders of pressure continuity m

$R = 100$ ft for all five distributed well models. However, Figure 5-8b shows that the pressure profiles produced by the $m = 1$ and $m = 2$ distributed well models have discontinuities in the third derivative across $r = R$, whereas the profiles for $m \geq 3$ remain continuous. This behavior is expected since an order m distributed well produces pressure profiles that are $\mathcal{C}^m$-continuous.

5.2.3 Extension to anisotropic permeability

This subsection extends the proposed distributed well model to problems with diagonally anisotropic absolute permeability tensors, similar to Peaceman's extension discussed in Section 5.1.1. The key idea behind this extension is to find a coordinate transformation which transforms the anisotropic problem into an isotropic problem, which can then be solved using the formulation given in Section 5.2.2.

Consider the following coordinate transformation from elliptic cylindrical coordinates $(\tilde{r}, \tilde{\theta}, \tilde{z})$ to Cartesian coordinates (x, y, z),

$$\begin{aligned} x &= \frac{1}{2}\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}}\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)\tilde{r}\cos(\tilde{\theta}), \\ y &= \frac{1}{2}\left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)\tilde{r}\sin(\tilde{\theta}), \\ z &= \tilde{z}, \end{aligned} \tag{5.40}$$

where k_{xx} and k_{yy} are the horizontal components of the diagonal permeability tensor $\mathbf{K} = \text{diag}(k_{xx}, k_{yy}, k_{zz})$ and curves of constant $\tilde{r}$ form ellipses. The above transformations can be differentiated and used with chain rules to show that,

$$k_{xx}\frac{\partial^2 p}{\partial x^2} + k_{yy}\frac{\partial^2 p}{\partial y^2} + k_{zz}\frac{\partial^2 p}{\partial z^2} = \frac{\sqrt{k_{xx}k_{yy}}}{|J|}\left(\frac{\partial p}{\partial \tilde{r}} + \tilde{r}\frac{\partial^2 p}{\partial \tilde{r}^2} + \frac{1}{\tilde{r}}\frac{\partial^2 p}{\partial \theta^2}\right) + k_{zz}\frac{\partial^2 p}{\partial \tilde{z}^2}., \tag{5.41}$$

where $|J|$ is the determinant of the Jacobian matrix of the coordinate transform, given by,

$$|J| = \left|\frac{\partial(x, y, z)}{\partial(\tilde{r}, \tilde{\theta}, \tilde{z})}\right| = \frac{1}{4}\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)^2 \tilde{r}. \tag{5.42}$$

The pressure equation for the anisotropic problem is given by,

$$-\frac{1}{\mu}\left(k_{xx}\frac{\partial^2 p}{\partial x^2} + k_{yy}\frac{\partial^2 p}{\partial y^2} + k_{zz}\frac{\partial^2 p}{\partial z^2}\right) = \tilde{q}(x, y), \tag{5.43}$$

where $\tilde{q}(x, y)$ is the distributed source term required for the anisotropic problem. Applying the coordinate transform by substituting Eq. 5.41 into the above equation yields,

$$-\frac{1}{\mu}\left(\frac{\sqrt{k_{xx}k_{yy}}}{|J|}\left(\frac{\partial p}{\partial \tilde{r}} + \tilde{r}\frac{\partial^2 p}{\partial \tilde{r}^2} + \frac{1}{\tilde{r}}\frac{\partial^2 p}{\partial \tilde{\theta}^2}\right) + k_{zz}\frac{\partial^2 p}{\partial \tilde{z}^2}\right) = \tilde{q}(\tilde{r}, \tilde{\theta}). \tag{5.44}$$

Assuming that the pressure is independent of $\tilde{\theta}$ and $\tilde{z}$ simplifies the above equation to,

$$
\begin{aligned}
-\frac{\sqrt{k_{xx}k_{yy}}}{\mu}\left(\frac{\partial p}{\partial \tilde{r}} + \tilde{r}\frac{\partial^2 p}{\partial \tilde{r}^2}\right) &= \tilde{q}(\tilde{r})\,|J|\,, \\
-\frac{\sqrt{k_{xx}k_{yy}}}{\mu}\left(\frac{1}{\tilde{r}}\frac{\partial p}{\partial \tilde{r}} + \frac{\partial^2 p}{\partial \tilde{r}^2}\right) &= \tilde{q}(\tilde{r})\;\frac{1}{4}\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)^2 . \qquad (5.45)
\end{aligned}
$$

Furthermore, rewriting Peaceman's analytic solution in Eq. 5.13 in the elliptic cylindrical coordinates yields,

$$
\begin{aligned}
p_{\text{analytic}} &= p_{wf} - \frac{\mu Q}{2\pi\sqrt{k_{xx}k_{yy}}L_z}\ln\left(\frac{\sqrt{\left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{2}}x^2 + \left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{2}}y^2}}{\frac{1}{2}r_w\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)}\right), \\
&= p_{wf} - \frac{\mu Q}{2\pi\sqrt{k_{xx}k_{yy}}L_z}\ln\left(\frac{\tilde{r}}{r_w}\right). \qquad (5.46)
\end{aligned}
$$

Comparing the above equation with Eq. 5.5 shows that the anisotropic problem written in the elliptic cylindrical coordinates is equivalent to an isotropic problem with an effective permeability of $K = \sqrt{k_{xx}k_{yy}}$. Hence, the source term required for the anisotropic problem can be of the same form as the distributed source term derived for the isotropic problem in Section 5.2.2, with the elliptic radius $\tilde{r}$ being used instead of the usual cylindrical radius r. In particular, comparing the source terms of Eqs. 5.45 and 5.21 shows that,

$$
\tilde{q}(\tilde{r}) = q(\tilde{r})\cdot 4\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)^{-2}, \qquad (5.47)
$$

where $q(\tilde{r})$ is the source term derived for the isotropic problem and $\tilde{q}(\tilde{r})$ is the modified source term required for the anisotropic problem. Using $q(\tilde{r})$ from Eq. 5.26, the final form of $\tilde{q}(\tilde{r})$ is given by,

$$
\boxed{\tilde{q}(\tilde{r}) = \begin{cases} -\frac{\tilde{K}}{\mu}\left(\frac{p(\tilde{r})-p_{wf}}{1-f(\tilde{s})-f'(1)\ln\left(\frac{R}{r_w}\right)}\right)\frac{1}{R^2}\left(-\frac{1}{\tilde{s}}f'(\tilde{s}) - f''(\tilde{s})\right), & \tilde{r} \le R, \\ 0, & \tilde{r} > R, \end{cases}} \qquad (5.48)
$$

where,

$$\tilde{K} = \frac{4\sqrt{k_{xx}k_{yy}}}{\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)^2}, \tag{5.49}$$

and $\tilde{r}$ is the elliptic radius from Eq. 5.40 rewritten as,

$$\tilde{r} = \frac{\left(\sqrt{\frac{k_{yy}}{k_{xx}}}x^2 + \sqrt{\frac{k_{xx}}{k_{yy}}}y^2\right)^{\frac{1}{2}}}{\frac{1}{2}\left(\left(\frac{k_{xx}}{k_{yy}}\right)^{\frac{1}{4}} + \left(\frac{k_{yy}}{k_{xx}}\right)^{\frac{1}{4}}\right)}, \tag{5.50}$$

and $\tilde{s}$ is defined as,

$$\tilde{s} = \frac{\tilde{r}}{R}. \tag{5.51}$$

Note that for the special case $k_{xx} = k_{yy} = K$, the coordinate transformation in Eq. 5.40 reduces to the standard cylindrical coordinates, and the source term $\tilde{q}(\tilde{r})$ reduces to $q(r)$. All subsequent numerical results obtained using the distributed well model in this thesis use the source term defined above in Eqs. 5.48 - 5.51.

5.2.4 Extension to multi-phase flow

The distributed well model derived for single-phase flow in Sections 5.2.2 - 5.2.3 can also be extended to multi-phase flows by weighting the source term $\tilde{q}(\tilde{r})$ by the relative mobility of each phase. The source term for phase α is then given by,

$$\tilde{q}_\alpha(\tilde{r}) = \begin{cases} -\frac{k_{r\alpha}}{\mu_\alpha}\tilde{K}\left(\frac{p(\tilde{r})-p_{wf}}{1-f(\tilde{s})-f'(1)\ln\left(\frac{R}{r_w}\right)}\right)\frac{1}{R^2}\left(-\frac{1}{\tilde{s}}f'(\tilde{s}) - f''(\tilde{s})\right), & \tilde{r} \leq R, \\ 0, & \tilde{r} > R, \end{cases} \tag{5.52}$$

where μ_α represents the phase viscosity and $k_{r\alpha}$ is the relative permeability function of phase α. If the well is an injection well, the relative permeability is evaluated from the saturation of the injected fluid. If the well is a producer, the relative permeability is evaluated from the saturation of the outgoing fluid.

5.3 Numerical results

This section presents numerical results for a steady, single-phase flow problem with a single well, and an unsteady two-phase flow problem with two wells. These problems are solved with a finite volume method using a two-point flux approximation (TPFA) on uniform structured quadrilateral meshes, with both Peaceman's well model (Eq. 5.16) and the proposed distributed well model (Eq. 5.48). Further, the finite volume results are also compared with results from a high-order discontinuous Galerkin method using the distributed well model. In this work, the DG method uses the second formulation proposed by Bassi and Rebay (BR2) [21, 22] to discretize diffusive fluxes. The DG results are obtained on uniform quadrilateral meshes and also on unstructured simplex meshes produced by the output-based mesh adaptation framework described in Chapter 2.

5.3.1 Steady single-phase flow problem

The objective of this problem is to ensure that the proposed distributed well model demonstrates a pressure - flow rate relationship that is consistent with the analytic relationship assumed in Peaceman's work (Eq. 5.13). This problem considers a homogeneous reservoir in a square domain $\Omega_s = [-L, L] \times [-L, L]$, where $L = 1000$ ft, with a single production well located at the origin. The depth of the reservoir is not explicitly modeled due to the assumed uniformity of the solutions in the z-direction, thus reducing the 3D problem to a 2D problem (i.e. $L_z = 1$). The absolute permeability field is assumed to be diagonally anisotropic. Since an infinite domain cannot be solved numerically, the problem is posed in reverse where the pressure on the four boundaries of the square domain is set using $p_{\text{analytic}}(x, y)$ for a specified flow rate Q_{spec}, and the well models are evaluated on how closely their predicted flow rates match Q_{spec}. The governing PDE for the fluid pressure $p(x, y)$ is given by,

$$-\frac{1}{\mu}\left(k_{xx}\frac{\partial^2 p}{\partial x^2} + k_{yy}\frac{\partial^2 p}{\partial y^2}\right) = q(x, y), \qquad \forall (x, y) \in \Omega_s, \tag{5.53}$$

with the Dirichlet boundary condition,

$$p(x,y) = p_{\text{analytic}}(x,y), \qquad \forall (x,y) \in \partial\Omega_s, \tag{5.54}$$

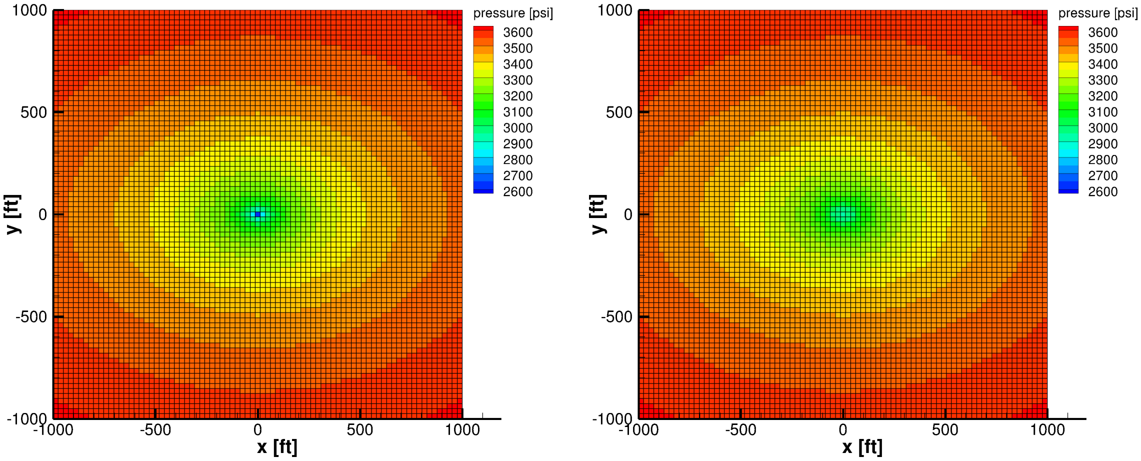
where $p_{\text{analytic}}(x,y)$ is evaluated using Eq. 5.13 with a flow rate Q_{spec}. The relevant parameters for the problem are listed below,

$$\begin{aligned} k_{xx} &= 200 \text{ mD}, & k_{yy} &= 100 \text{ mD}, \\ \mu &= 1 \text{ cP}, & Q_{\text{spec}} &= -1000 \text{ ft}^3/\text{day}, \\ p_{wf} &= 2000 \text{ psi}, & r_w &= 2 \text{ inches}. \end{aligned}$$

The source term $q(x,y)$ is modeled using either Peaceman's well model or the distributed well model, and results of both cases are presented below. Figure 5-9 shows contour plots of pressure obtained using a finite volume method with a two-point flux approximation on a structured quadrilateral mesh. Figures 5-9a and 5-9b contain the results of Peaceman's well model and a distributed well model with sixth-order pressure continuity ($m = 6$) respectively. Figure 5-10 contains a plot of the pressure distribution produced by a piecewise linear (P1) discontinuous Galerkin method, on the same quadrilateral mesh. Finally, Figures 5-11a and 5-11b show piecewise linear (P1) and piecewise quadratic (P2) DG solutions on unstructured triangle meshes, which were adapted to minimize the error in the total volumetric flow rate, subject to a cost constraint of 10,000 degrees of freedom (DOF) in each solution. Both adapted solutions shown use distributed wells with $m = 6$ and a model radius of $R = 100$ ft, and were obtained after 20 iterations of the output-based mesh adaptation algorithm given in Chapter 2.

The pressure distribution outside the radius $r = R = 100$ ft appears identical in Figures 5-9 - 5-11, and this is further verified by Figure 5-12 which contains line plots of the pressure along the $x = y$ line. Figure 5-12 shows that the discrete pressure profiles, from both Peaceman's well model and the distributed well model, agree well with the analytic pressure profile (dashed black line) given by Eq. 5.13 in the region

$r > R$. When the well block size is roughly equal to the diameter of the modeled region of the distributed well ($2R$), the well block pressure predicted by Peaceman's well model is nearly identical to that of the distributed well model (black and brown dotted lines).



(a) Peaceman's well model

(b) Distributed well model with $m = 6$ and $R = 100$ ft

Figure 5-9: Pressure solutions obtained with a finite volume method on a 81 × 81 quadrilateral mesh

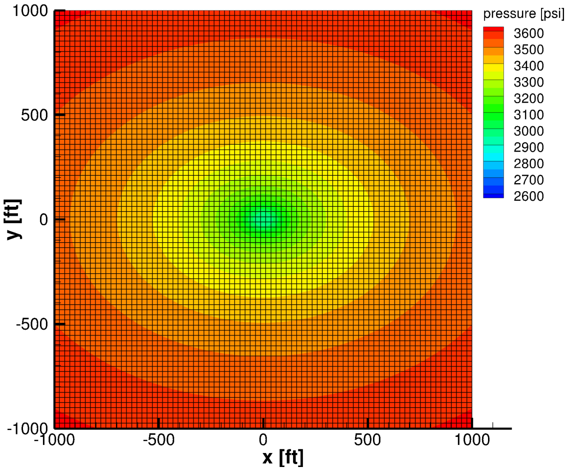
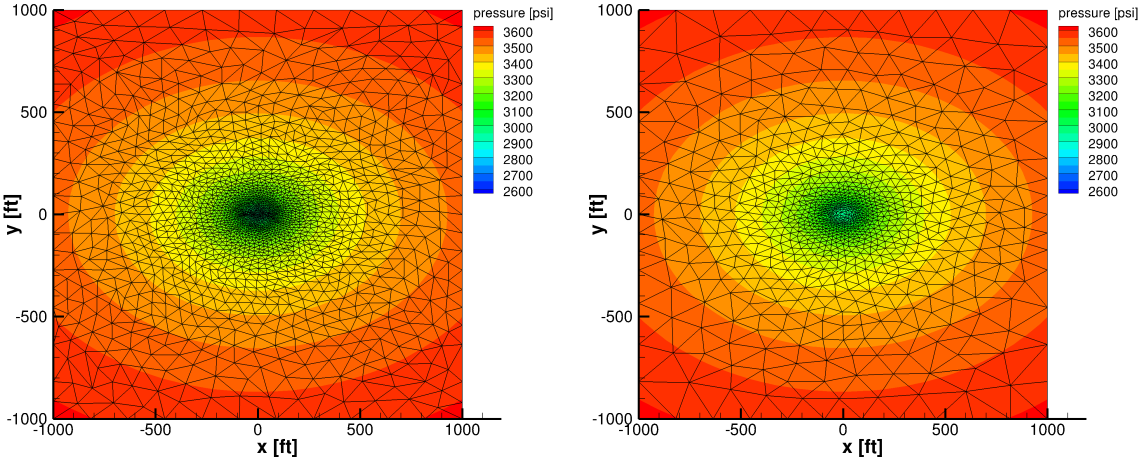


Figure 5-10: Pressure solution obtained with a piecewise linear (P1) discontinuous Galerkin method on a 81 × 81 quadrilateral mesh, using the distributed well model with $m = 6$ and $R = 100$ ft

The volumetric flow rate predictions for the finite volume method with Peaceman's well model are obtained using Eq. 5.16, whereas the predictions of the distributed



(a) Piecewise linear adapted solution (P1)

(b) Piecewise quadratic adapted solution (P2)

Figure 5-11: Pressure solutions obtained with a discontinuous Galerkin method on simplex meshes adapted to 10,000 degrees of freedom, using the distributed well model with $m = 6$ and $R = 100$ ft

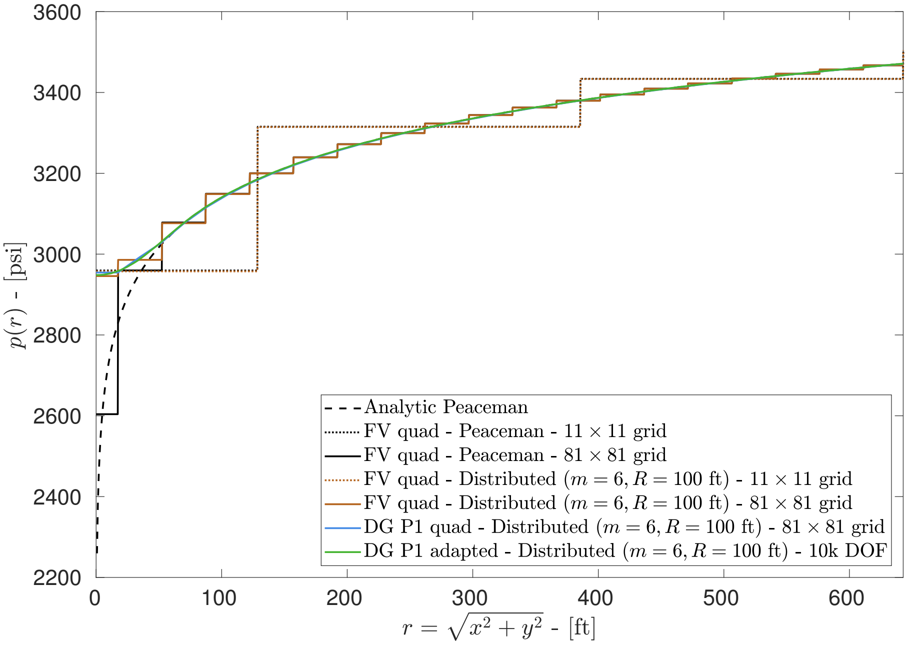


Figure 5-12: Comparison of discrete pressure profiles along $x = y$ with Peaceman's analytic solution

well model are obtained by numerically integrating the distributed source term $q(r)$ over the domain. The accuracy of these flow rate predictions rates is evaluated by comparing with the true volumetric flow rate Q_{spec}. Figure 5-13 shows how the

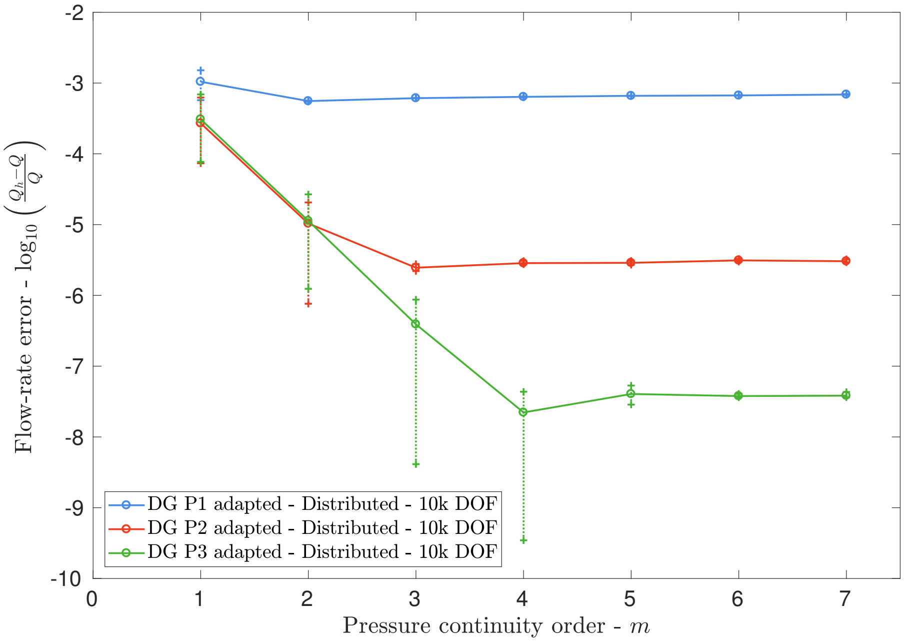
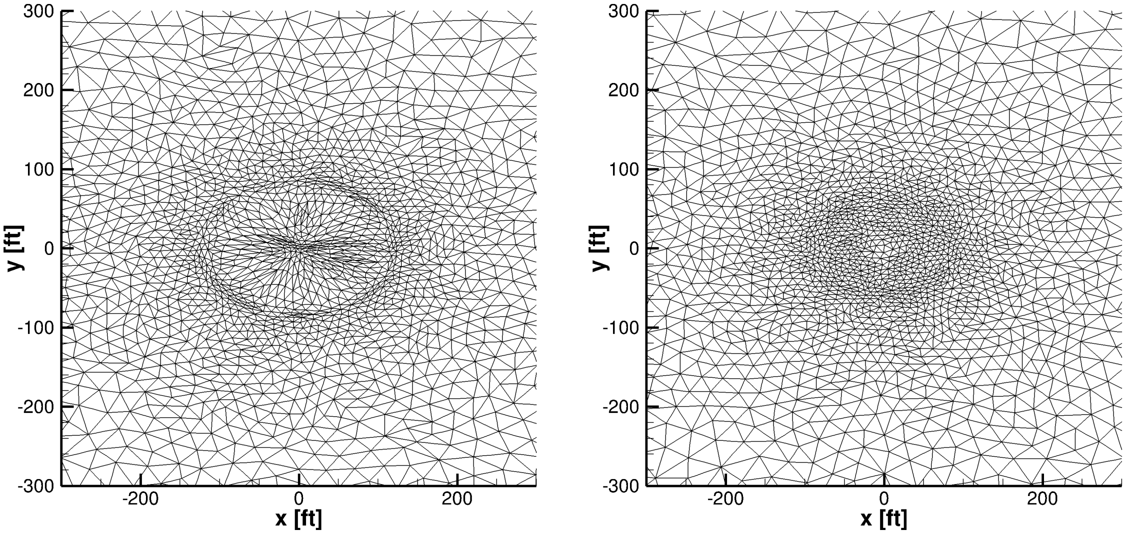


Figure 5-13: Volumetric flow rate errors for distributed well models with different continuity orders

relative error in the predicted volumetric flow rate varies with the order of pressure continuity of the distributed well model. The solid blue, red and green lines represent piecewise linear (P1), piecewise quadratic (P2) and piecewise cubic (P3) DG solutions respectively, on unstructured triangle meshes adapted to 10,000 degrees of freedom over 20 mesh adaptation iterations. The circular markers plot the average flow rate error of the last 5 mesh adaptation iterations, and the vertical error bars represent the range or the “spread” of flow rate errors over the last 5 adapted meshes. It is clearly observed that the smoothness of the pressure profile greatly impacts the accuracy of the output, and higher-order discretizations require distributed well models with larger m to consistently predict flow rates with smaller errors.

Figure 5-14 highlights the impact of the smoothness of the distributed well on the final meshes produced by the mesh adaptation algorithm. Figure 5-14a shows the final mesh obtained using a distributed well with $m = 2$, which has an analytic pressure profile that is continuous up to the second derivative. Similarly, Figure 5-14b shows the final mesh for a distributed well with $m = 6$, which has an analytic pressure profile that is continuous up to the sixth derivative. Both problems are solved using a piecewise cubic (P3) DG discretization. The ring-like adaptation feature seen around the active region of the well in Figure 5-14a is a result of the adaptive algorithm trying to accurately capture the discontinuity in the third derivative of pressure across the boundary of the well, which is "seen" by the P3 DG solution. However, using a distributed well model with $m = 6$ shifts the discontinuity in the analytic pressure profile to the seventh derivative, thereby making the discontinuity invisible to a P3 DG solution. As seen in Figure 5-14b, this eliminates the need for additional mesh resolution along the boundary of the well, and instead allows for increased resolution inside the well. Since this work only considers high-order discretizations up to P3, a pressure continuity order of $m = 6$ was deemed appropriate based on Figure 5-13. Therefore, all subsequent distributed well models in this work use the $f_6(s)$ activation function given in Table 5.1.



(a) Distributed well model with $m = 2, R =$ 100 ft

(b) Distributed well model with $m = 6, R =$ 100 ft

Figure 5-14: Zoomed-in plots of the final adapted meshes for piecewise cubic (P3) DG solutions with 50,000 degrees of freedom, using the distributed well model

Figure 5-15 shows how the relative error in the volumetric flow rate behaves with an average mesh size h, for different discretizations. In order to facilitate a fair comparison between solutions of different orders, on both structured and unstructured meshes, the average mesh size h is taken to be inversely proportional to the square root of the total number of degrees of freedom N in each solution. The dashed black line shows results from Peaceman's well model, whereas all other lines represent results from a distributed well with $m = 6$ and $R = 100$ ft. The finite volume results (dashed lines) exhibit a second-order convergence rate, which agrees with the accuracy of the FV scheme for elliptic problems. The P1, P2 and P3 DG finite element results (solid lines) on structured and adapted meshes show second-, fourth- and sixth-order convergence rates respectively, thereby confirming the expected output superconvergence rate of $\mathcal{O}(h^{2P})$. Peaceman's well model is observed to work exceptionally for the situation it was designed for, which in essence, allows the behavior of the well to be modeled in a single grid block. It is also worth noting that Peaceman's well model is invalid in the grey region of the plot, where the h values are sufficiently small to violate the $r_e > r_w$ constraint. In contrast, the distributed well model allows for arbitrarily small h.

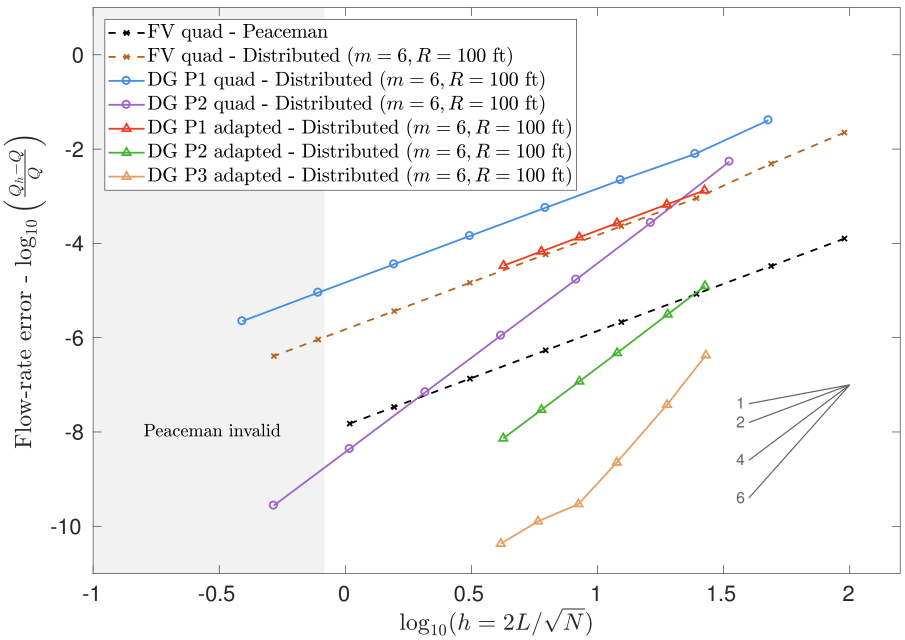


Figure 5-15: Volumetric flow rate error vs. average mesh size for different discretizations

The effect of varying the size of the modeled region of the well is shown in Figure 5-16, where the flow rate error convergence behavior of Peaceman's well model is compared to those of distributed wells with a fixed continuity order of $m = 6$

and different model radii. The plot only shows results obtained on uniformly refined quadrilateral meshes, using both FV and DG discretizations. The finite volume and DG P1 results exhibit a second-order convergence rate while the DG P2 (solid purple lines) results show fourth-order convergence as expected. It is clearly visible that increasing the model radius R of the distributed well produces a lower error, for the same convergence rate. Furthermore, if the distributed well results are plotted against an abscissa of h/R instead of h, as done in Figure 5-17, the curves for different R collapse to a single line for each discretization. This implies that the output error depends explicitly on the ratio h/R, instead of h and R separately.

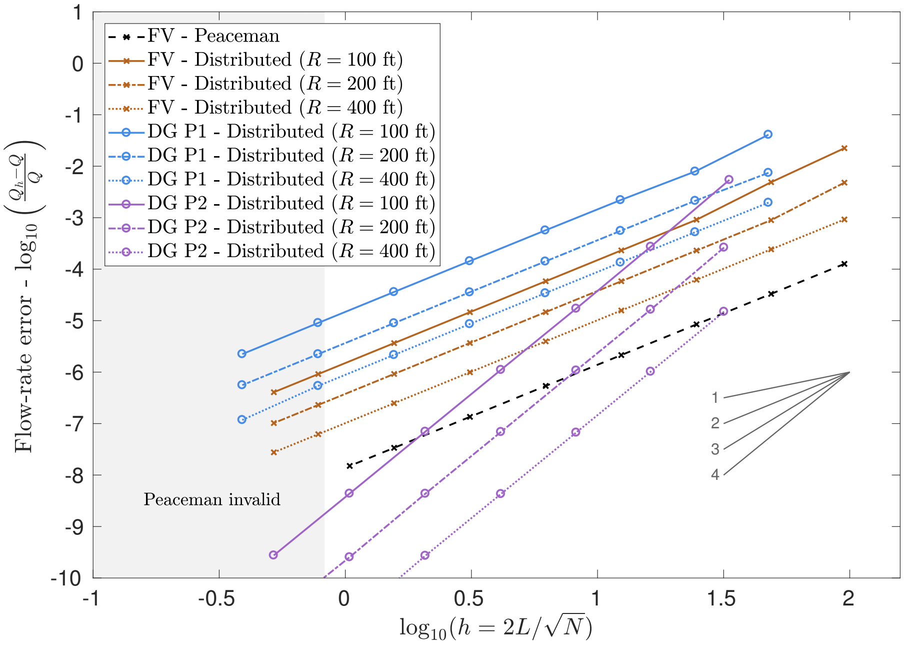


Figure 5-16: Volumetric flow rate errors for Peaceman's well model and the distributed well model with $m = 6$ and different model radii

The sensitivity of the well models to the nature of the mesh is investigated in Figure 5-18, which shows the results of Peaceman's well model and the single-parameter distributed well using the finite volume method on four different families of rectangular grids. The black lines represent results on uniformly refined square meshes with constant spacing, whereas the blue, red and green lines represent results on non-uniform rectangular meshes generated using Eqs. 5.18 and 5.19, with β values of 0.01, 0.1 and 1, respectively. Peaceman's well model outperforms the distributed

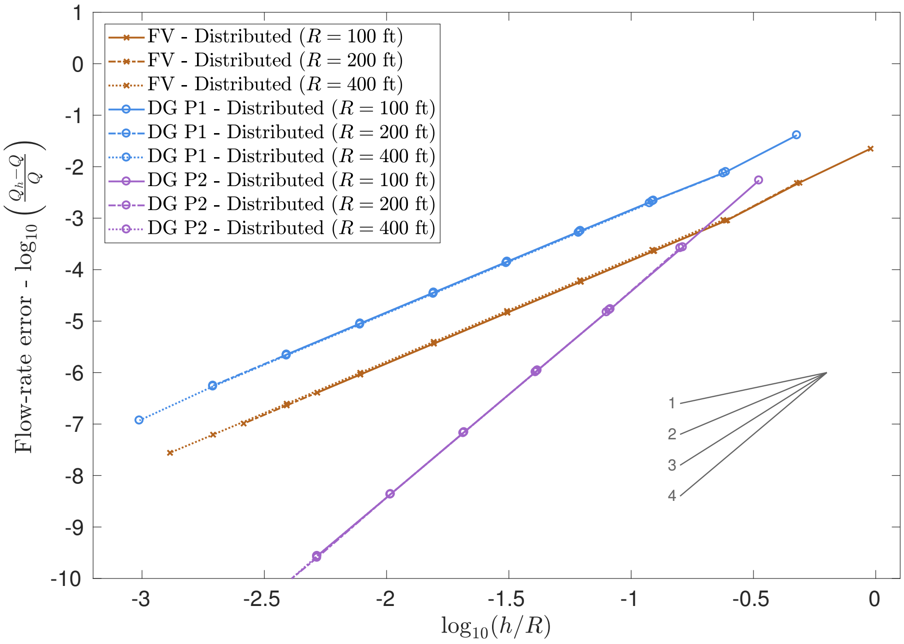


Figure 5-17: Volumetric flow rate error vs h/R for distributed well models with different model radii

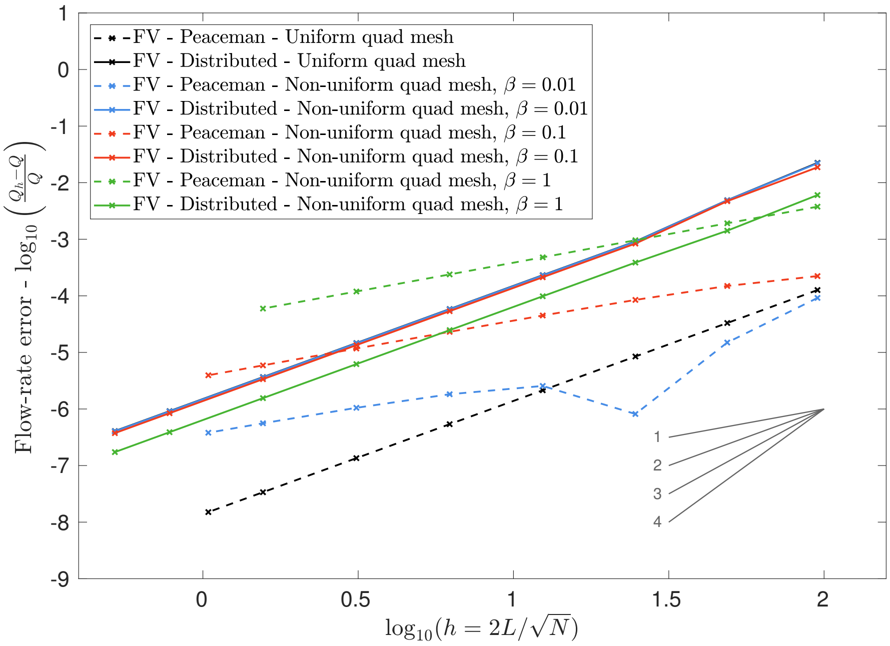


Figure 5-18: Volumetric flow rate errors on uniform and non-uniform rectangular meshes, with Peaceman's well model and a distributed well model with $m = 6$, $R = 100$ ft

well model on the uniform meshes as seen before, but fails to achieve second-order convergence on the slightly non-uniform meshes. This poor convergence behavior is due to the violation of the *uniform* mesh assumption used in the derivation of the equivalent radius r_e in Peaceman's model. Although some of the non-uniform meshes are virtually indistinguishable from the uniform meshes (i.e. $\beta = 0.01$), the sensitivity of r_e to the mesh is significant enough to cause Peaceman's well model to lose second-order convergence. It can be argued that the reason Peaceman's well model performs exceptionally well only on uniform rectangular meshes is because it is, in effect, fine-tuned to that particular class of meshes through the definition of r_e. In contrast, the discretization-agnostic nature of the distributed well model (solid lines) enables it to achieve a robust second-order convergence with similar error levels across all four mesh families.

The results presented in this section clearly show that the proposed distributed well model produces pressure profiles and flow rates that are consistent with the analytic relationship derived by Peaceman. In terms of predicting the flow rate accurately, Peaceman's well model outperforms the distributed well model with the finite volume method on uniform rectangular meshes. However, the generality of the distributed well model enables it to be used on non-uniform structured and unstructured meshes as well as with any discretization without any modifications.

5.3.2 Two-phase flow problem

The unsteady, incompressible two-phase flow problem considered in this section involves a two-spot well configuration in a square domain $\Omega_s = [0, L] \times [0, L]$, where $L = 3000$ ft. As before, the depth of the reservoir is assumed to be unity ($L_z = 1$) and is not explicitly modeled, resulting in a 2D spatial problem. As shown by the schematic in Figure 5-19, the problem consists of an injector and a producer located at coordinates $(500, 500)$ and $(2500, 2500)$ respectively. The injection and production wells have fixed bottom-hole pressures of 4000 psi and 2000 psi respectively.

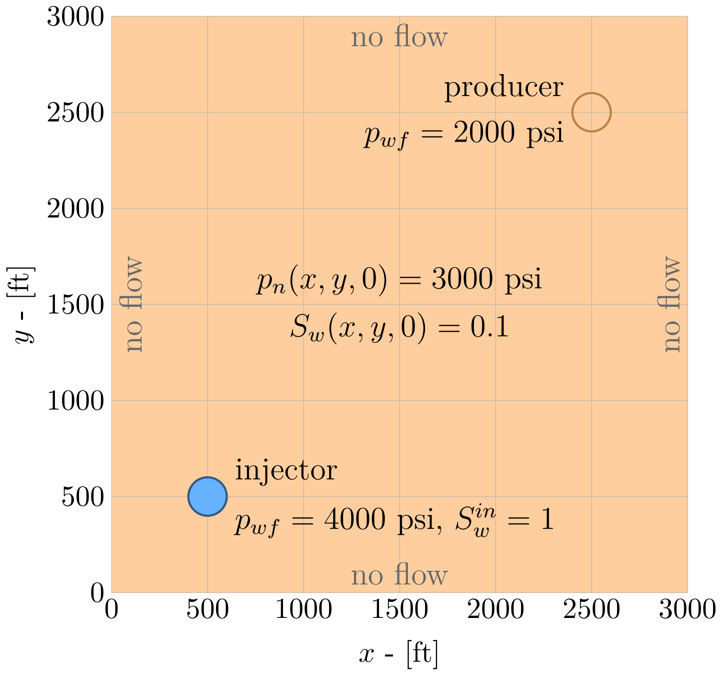


Figure 5-19: Schematic of reservoir

The governing equations for this problem are given by Eq. 5.55 below,

$$\begin{aligned} \frac{\partial}{\partial t}\left(\rho_w \phi S_w\right) - \nabla \cdot \left(\rho_w \frac{k_{rw}}{\mu_w} \mathbf{K} \nabla p_w\right) &= \rho_w q_w \\ \frac{\partial}{\partial t}\left(\rho_n \phi S_n\right) - \nabla \cdot \left(\rho_n \frac{k_{rn}}{\mu_n} \mathbf{K} \nabla p_n\right) &= \rho_n q_n, \qquad \forall (x, y) \in \Omega_s, \end{aligned} \tag{5.55}$$

where ρ_w and ρ_n are the phase fluid densities (with subscripts w and n denoting the wetting and non-wetting phases, respectively), ϕ is the rock porosity, $\mathbf{K}$ is the absolute permeability tensor, k_{rw} and k_{rn} are the relative permeability functions, and μ_w and μ_n are the fluid viscosities. The behavior of the wells is modeled using the source terms q_w and q_n, which use the distributed well model formulation given in Eq. 5.52, or the equivalent multi-phase extension of Peaceman's well model. The phase pressures and saturations are given by p_w, p_n and S_w, S_n respectively, which are related via the closure equations,

$$\begin{aligned} p_w + p_c &= p_n, \\ S_w + S_n &= 1, \end{aligned} \tag{5.56}$$

where p_c is the capillary pressure. The primary variables $p_n(x, y, t)$ and $S_w(x, y, t)$ have the following initial conditions,

$$\begin{aligned} p_n(x, y, 0) &= 3000 \text{ psi} \\ S_w(x, y, 0) &= 0.1. \end{aligned} \tag{5.57}$$

The boundary conditions represent no flow conditions on all four boundaries of the domain, written as,

$$\begin{aligned} &\mathbf{K}\nabla p_w \cdot \vec{n} = 0, \\ &\mathbf{K}\nabla p_n \cdot \vec{n} = 0, \qquad \forall(x, y) \in \partial\Omega_s, \end{aligned} \tag{5.58}$$

where $\vec{n}$ is the outward pointing unit normal vector on each boundary. The constitutive relationships and parameters for this problem are given by,

$$\begin{aligned} k_{r\alpha} &= S_\alpha^2, \qquad && \text{for } \alpha \in \{w, n\} \\ p_c &= p_{c_{\max}}(1 - S_w), \end{aligned} \tag{5.59}$$

and,

$$\begin{aligned} \rho_w &= 62.4 \text{ lb/ft}^3, \qquad & \rho_n &= 52.1 \text{ lb/ft}^3, \\ \phi &= 0.3, & \mathbf{K} &= 200 \ \mathbf{I} \text{ mD}, \\ \mu_w &= 1 \text{ cP}, & \mu_n &= 2 \text{ cP}, \\ p_{c_{\max}} &= 5 \text{ psi}, & r_w &= 2 \text{ inches}. \end{aligned}$$

The objective of this problem is to compare the results of using Peaceman's well model with those of the proposed distributed well model across different numerical discretizations of Eqs. 5.55 - 5.58. Assuming a water-oil system, the output functional of interest is the oil recovery factor given by,

$$J = \frac{V_{n_{out}}}{V_{OIP}}, \tag{5.60}$$

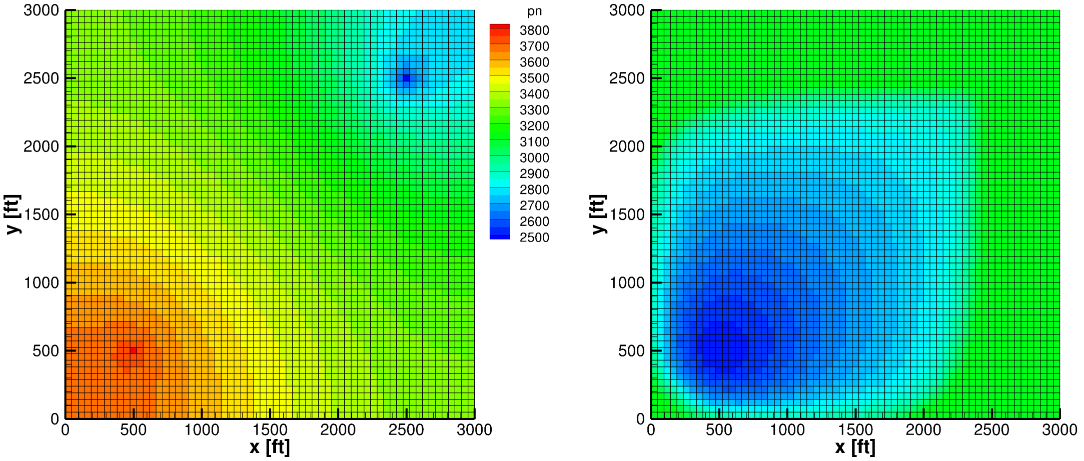
where $V_{n_{out}}$ is the total volume of oil extracted from the production well over a period of $T = 2500$ days, and V_{OIP} is the total volume of oil-in-place at $t = 0$, defined as follows,

$$V_{n_{out}} = L_z \int_0^T \int_0^L \int_0^L -q_n \; dx \; dy \; dt, \tag{5.61}$$

$$V_{OIP} = L_z \int_0^L \int_0^L \phi \cdot (1 - S_w(x, y, 0)) \; dx \; dy = 2.43 \times 10^6 \text{ ft}^3. \tag{5.62}$$

The two-phase flow problem described above is first solved with a finite volume method using a two-point flux approximation (TPFA) and a backward Euler (BDF1) time-marching scheme with a constant timestep, on structured quadrilateral meshes. Figures 5-20 and 5-21 show snapshots of the pressure and saturation solutions obtained from the finite volume method at $t = 2500$ days, for Peaceman's well model and the distributed well model respectively. The distributed well model uses a formulation with sixth-order pressure continuity ($m = 6$) and $R = 100$ ft. Similarly, Figure 5-22 shows the solutions from a piecewise linear (P1) DG method with a second-order backward difference (BDF2) time-marching scheme, on a structured quadrilateral mesh. The solutions from both well models are visually identical, except for the small difference in pressure observed inside the active radius of the distributed well.

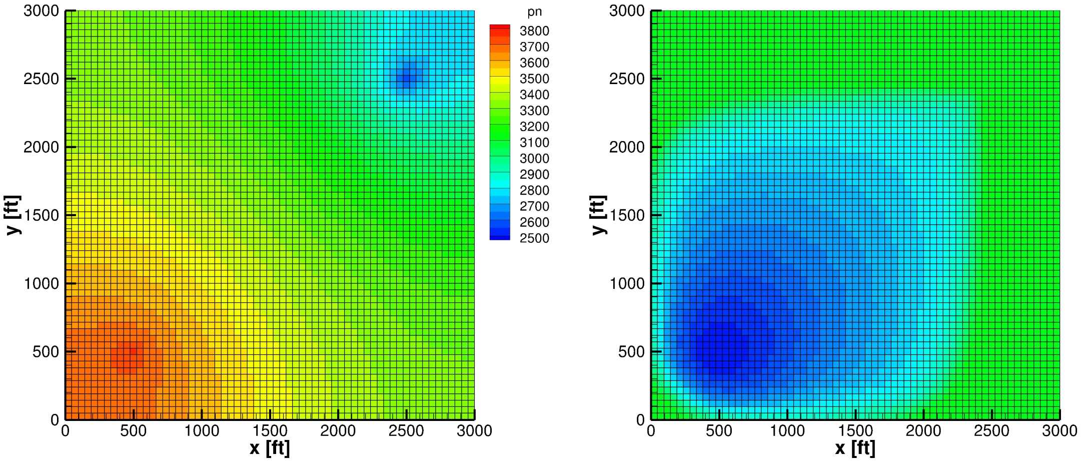
The water flooding problem is also solved using the adaptive space-time DG finite element method described in Chapter 2. The adaptive DG method is only used with the distributed well model, since Peaceman's well model is not readily applicable to finite element discretizations on fully unstructured meshes. Figure 5-23 shows the space-time pressure and saturation solutions obtained from the adaptive DG method with a piecewise linear (P1) approximation. In each sub-figure, the horizontal axes represent the spatial axes, and the vertical axis represents time. The solutions are represented on a fully unstructured, tetrahedral space-time mesh that is adapted to



(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 5-20: Solutions from a FV method with BDF1 using Peaceman's well model, on a 63 × 63 grid with 625 timesteps ($\Delta t = 4$ days), at $t = 2500$ days



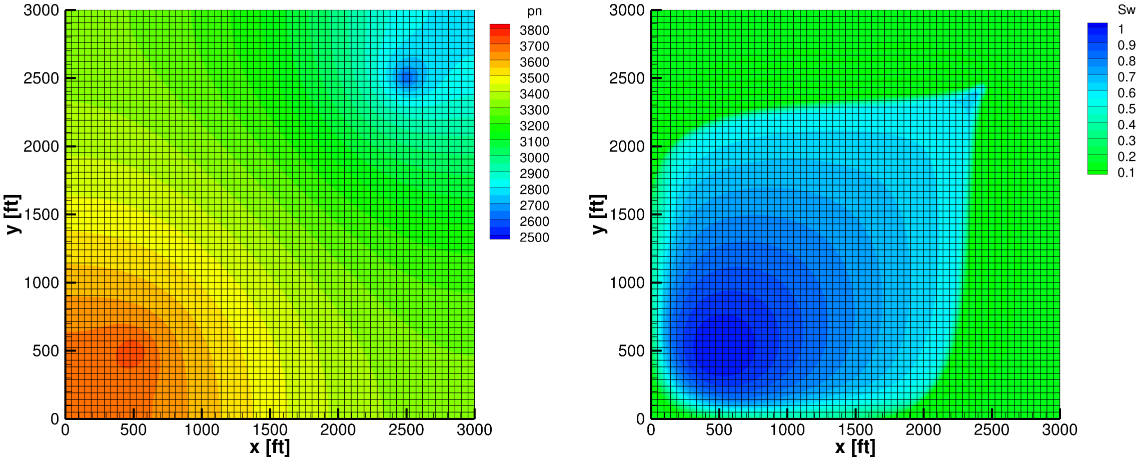
(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 5-21: Solutions from a FV method with BDF1 using a distributed well model with $m = 6$ and $R = 100$ ft, on a 63 × 63 grid with 625 timesteps ($\Delta t = 4$ days), at $t = 2500$ days

minimize the error in the output J, subject to a maximum cost constraint of 250,000 DOFs. It is clearly visible from the cross-sections in Figures 5-23c and 5-23d that the output-based mesh adaptation algorithm has allocated most of the DOFs to regions that have the greatest impact on the accuracy of the output, namely, the near-well regions and along the saturation front propagating through the spatial domain.

Figure 5-24 shows the convergence of oil recovery factors predicted by Peaceman's well model and a distributed well model with $m = 6$ and $R = 100$, using both FV



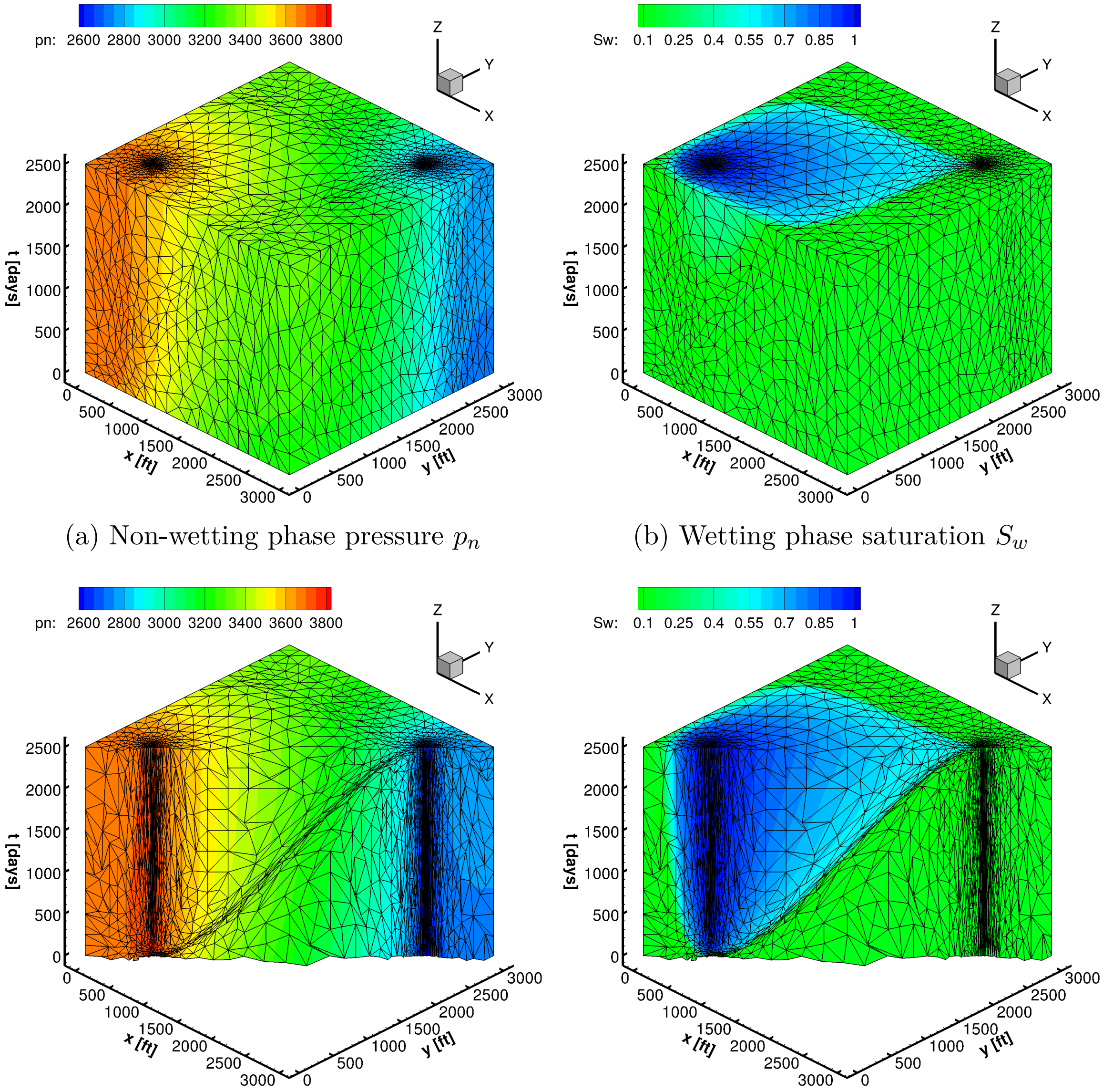
(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 5-22: Solutions from a piecewise linear (P1) DG method with BDF2 using a distributed well model with $m = 6$ and $R = 100$ ft, on a 63×63 grid with 625 timesteps ($\Delta t = 4$ days), at $t = 2500$ days

and DG discretizations. It is clearly seen that the space-time DG adapted methods (solid red and green lines) achieve the required error tolerance of $\pm 0.1\%$ (marked by the dotted horizontal lines) with about 4 orders of magnitude fewer space-time DOFs compared to the time-marching methods. Similarly, Figure 5-25 compares the output errors obtained from the two well models for different discretizations. The output error is defined as $\mathcal{E} = |J - J_h|$, where J_h is the discrete output for each case, and J is obtained from a high resolution reference solution computed using an adapted P2 DG solution containing 1 million DOFs. In order to fairly compare the results of time-marching and space-time discretizations, the non-dimensionalized average mesh size h is taken to be the cubic root of the number of space-time DOFs in each solution. The number next to a data point on the plot gives the total number of space-time DOFs in that discrete solution.

Unlike the results of the steady single-phase problem in Section 5.3.1, the finite volume results for Peaceman's well model and the distributed well model (dashed lines) show very similar levels of error in the output, for a given h. Both FV curves exhibit a first-order convergence in the output, where the output error is likely dominated by the error of the first-order time-marching scheme. The solid blue line represents solutions from the piecewise linear (P1) DG method with a BDF2 time-marching scheme,



(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

(c) Non-wetting phase pressure p_n (cross-section)

(d) Wetting phase saturation S_w (cross-section)

Figure 5-23: Piecewise linear (P1) solutions from the space-time adaptive DG method using a distributed well model with $m = 6$ and $R = 100$ ft, on a fully unstructured, tetrahedral space-time mesh adapted to 250,000 DOF

using the distributed well model. This method has a second-order accuracy in both space and time, and hence its output errors show a second-order convergence rate. Similarly, the solid purple line represents solutions from a piecewise quadratic (P2) DG method with a BDF3 time-marching scheme, where the output errors exhibit a third-order convergence rate. Lastly, the solid red and green lines represent the results of the adaptive space-time DG method with the distributed well model, for piecewise linear (P1) and piecewise quadratic (P2) solutions respectively. Fully unstructured mesh adaptation enables the space-time DG method to achieve superconvergence in the output, i.e. second- and fourth-order convergence for P1 and P2 respectively. Thus, it is capable of producing output values that are orders of magnitude more accurate for the same h, compared to the results of the time-marching schemes on structured meshes.

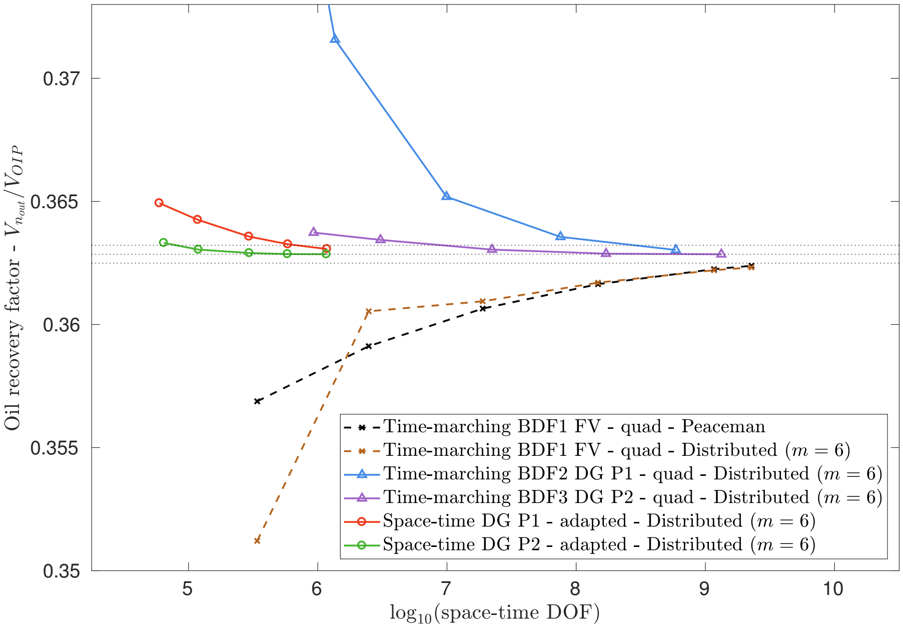


Figure 5-24: Output vs. space-time DOF plot for Peaceman's well model and the distributed well model with $m = 6$ and $R = 100$ ft

The results of this two-phase flow problem provide further evidence for the consistency of the distributed well model with the widely used Peaceman's well model. Further, the generality of the distributed well model makes it an attractive candidate for mesh adaptation frameworks and high-order discretizations, which allow the problem to be solved in a much more computationally efficient manner compared to conventional approaches.

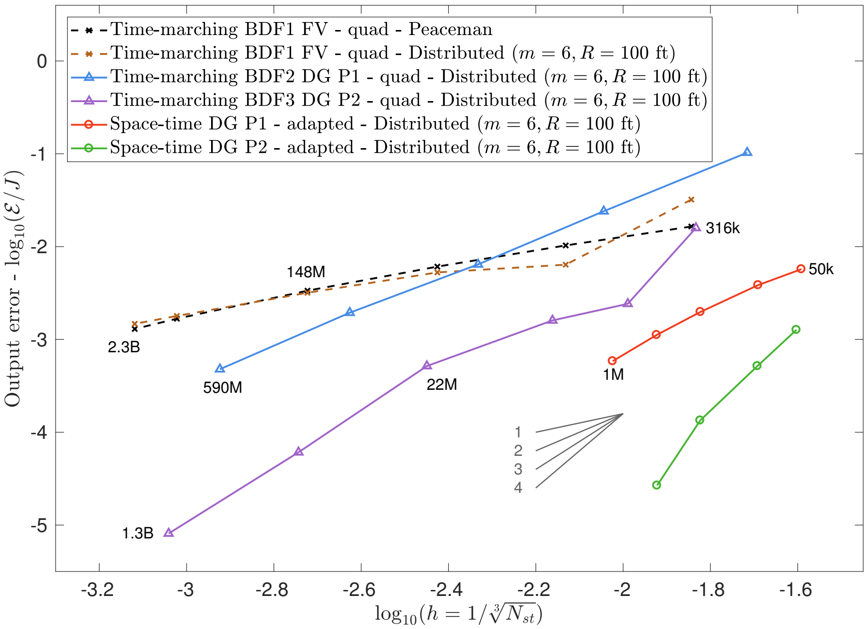


Figure 5-25: Volumetric flow rate errors for Peaceman's well model and the distributed well model with $m = 6$ and $R = 100$ ft

5.4 Summary

This chapter presented a distributed well model for describing the behavior of a fixed bottom-hole pressure well in reservoir simulation. The proposed well model uses a distributed, solution-dependent source term to mimic the behavior of a Dirichlet boundary condition that enforces the pressure at the well-bore radius to match the specified bottom-hole pressure. Unlike Peaceman-type well models, the distributed well model employs a discretization-independent formulation, which makes it directly applicable to the adaptive finite element framework used in this work. Furthermore, the presence of a user-specifiable model radius and benign pressure profiles inside the modeled region prevent the mesh adaptation algorithm from resolving down to the well-bore scale, thereby making the method feasible for realistic problems in large reservoirs. Extensions of the well model to anisotropic permeabilities and multi-phase flows were also presented. Results from grid convergence studies showed that the distributed well model produced volumetric flow rates that were consistent with Peaceman's model, while also being significantly more robust on non-uniform meshes.

Chapter 6

Compressible two-phase flow in a homogeneous reservoir

This chapter presents numerical results for a slightly compressible two-phase flow problem in a two-dimensional homogeneous reservoir. Output predictions from the adaptive space-time DG method are compared to those from conventional time-marching methods for computational efficiency.

6.1 Problem statement

The problem considered in this chapter is identical to the two-phase flow problem described in Section 5.3.2, except that the flow is slightly compressible in this case. As shown by the schematic in Figure 5-19, the problem consists of an injector and a producer located at coordinates $(500, 500)$ and $(2500, 2500)$ respectively, with fixed bottom-hole pressures of 4000 psi and 2000 psi respectively. The governing equations, boundary conditions and initial conditions are given by Eqs. 5.55 - 5.58, with the relevant constitutive relationships for the phase densities, porosity, relative permeabilities and capillary pressure given below,

$$\rho_\alpha = \rho_{\alpha_{\text{ref}}}\, e^{c_\alpha (p_\alpha - p_{\text{ref}})} \qquad \text{for } \alpha \in \{w, n\}, \tag{6.1}$$

$$\phi = \phi_{\text{ref}}\, e^{c_\phi (p_n - p_{\text{ref}})}, \tag{6.2}$$

$$k_{r\alpha} = S_\alpha^2, \qquad \text{for } \alpha \in \{w, n\} \tag{6.3}$$

$$p_c = p_{c_{\max}}(1 - S_w), \tag{6.4}$$

where,

$$\begin{aligned} \phi_{\text{ref}} &= 0.3, & p_{\text{ref}} &= 14.7 \text{ psi}, \\ \rho_{w_{\text{ref}}} &= 62.4 \text{ lb/ft}^3, & \rho_{n_{\text{ref}}} &= 52.1 \text{ lb/ft}^3, \\ c_w &= 5 \times 10^{-6} \text{ psi}^{-1}, & c_n &= 1.5 \times 10^{-5} \text{ psi}^{-1}, \\ c_\phi &= 3 \times 10^{-6} \text{ psi}^{-1}, & \mathbf{K} &= 200 \ \mathbf{I} \text{ mD}, \\ \mu_w &= 1 \text{ cP}, & \mu_n &= 2 \text{ cP}, \\ p_{c_{max}} &= 5 \text{ psi}, & r_w &= 2 \text{ inches}. \end{aligned}$$

The objective of this problem is to accurately predict the oil recovery factor given by,

$$J = \frac{V_{n_{out}}}{V_{OIP}}, \tag{6.5}$$

where $V_{n_{out}}$ is the total volume of oil extracted from the production well over a period of $T = 2500$ days, and V_{OIP} is the total volume of oil-in-place at $t = 0$, defined as follows,

$$V_{n_{out}} = L_z \int_0^T \int_0^L \int_0^L -q_n \ dx \ dy \ dt, \tag{6.6}$$

$$V_{OIP} = L_z \int_0^L \int_0^L \phi \cdot (1 - S_w(x, y, 0)) \ dx \ dy = 2.4519 \times 10^6 \text{ ft}^3. \tag{6.7}$$

6.2 Numerical results

The compressible two-phase flow problem described in the previous section is solved using the different discretizations listed below, and their output predictions are compared for accuracy and efficiency.

1. Finite volume (FV) method using a two-point flux approximation with upstream mobility weighting, and a backward Euler (BDF1) time-marching scheme. Solved

on uniformly spaced quadrilateral meshes. This method is representative of current industrial practices.

2. Piecewise linear (P1) DG method using a second order backward differentiation formula (BDF2) time-marching scheme. Solved on uniformly spaced quadrilateral meshes.

3. Piecewise quadratic (P2) DG method using a third order backward differentiation formula (BDF3) time-marching scheme. Solved on uniformly spaced quadrilateral meshes.

4. Piecewise linear (P1) space-time DG method on structured hexahedral meshes.

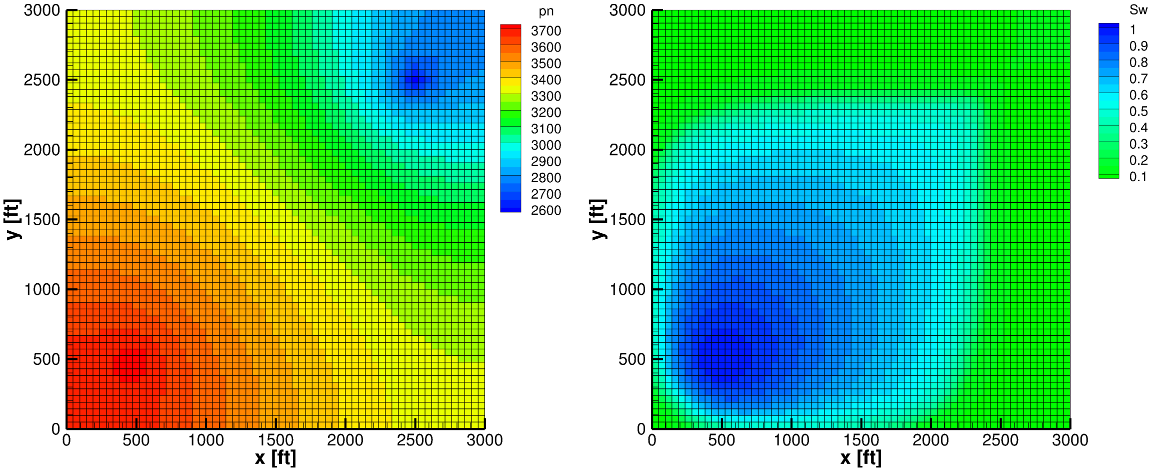
5. Piecewise quadratic (P2) space-time DG method on structured hexahedral meshes.

6. Piecewise linear (P1) adaptive space-time DG method on unstructured tetrahedral meshes.

7. Piecewise quadratic (P2) adaptive space-time DG method on unstructured tetrahedral meshes.

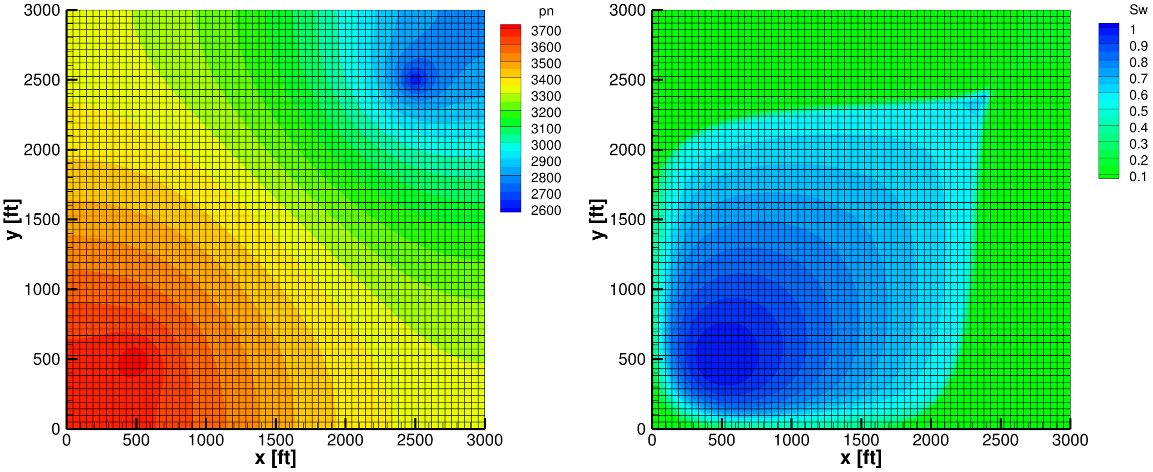
All of the DG discretizations mentioned above use the additional upwinding terms from Chapter 4. Furthermore, all seven methods use the distributed well model introduced in Chapter 5 for specifying the well behavior, with continuity order $m = 6$ and model radius $R = 100$ ft.

Figures 6-1 and 6-2 show snapshots of the pressure and saturation solutions, obtained from the finite volume method with BDF1 and the P1 DG method with BDF2 respectively. These solutions are very similar to the ones presented in Section 5.3.2, since the addition of compressibility effects does not change the overall flow behavior significantly. Figure 6-3 shows the final space-time mesh obtained after 20 mesh adaptation iterations, where the P1 DG solutions are adapted to a target of 10^6 degrees-of-freedom per primary variable. A crinkled cross-section of the final adapted mesh along the $x = y$ plane is given in Figure 6-4, together with the primal solutions which show the propagation of the water saturation front across the spatial domain.



(a) Non-wetting phase pressure p_n (b) Wetting phase saturation S_w

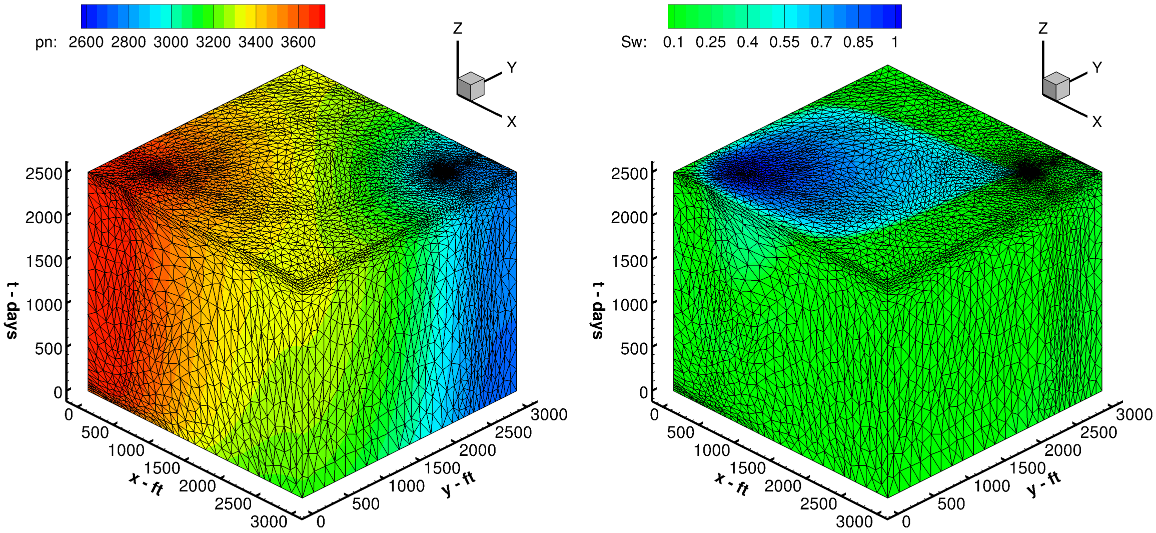
Figure 6-1: Primal solutions from the FV method with BDF1 on a 63 × 63 grid with 625 timesteps ($\Delta t = 4$ days), at $t = 2500$ days



(a) Non-wetting phase pressure p_n (b) Wetting phase saturation S_w

Figure 6-2: Primal solutions from the P1 DG method with BDF2 on a 63 × 63 grid with 625 timesteps ($\Delta t = 4$ days), at $t = 2500$ days

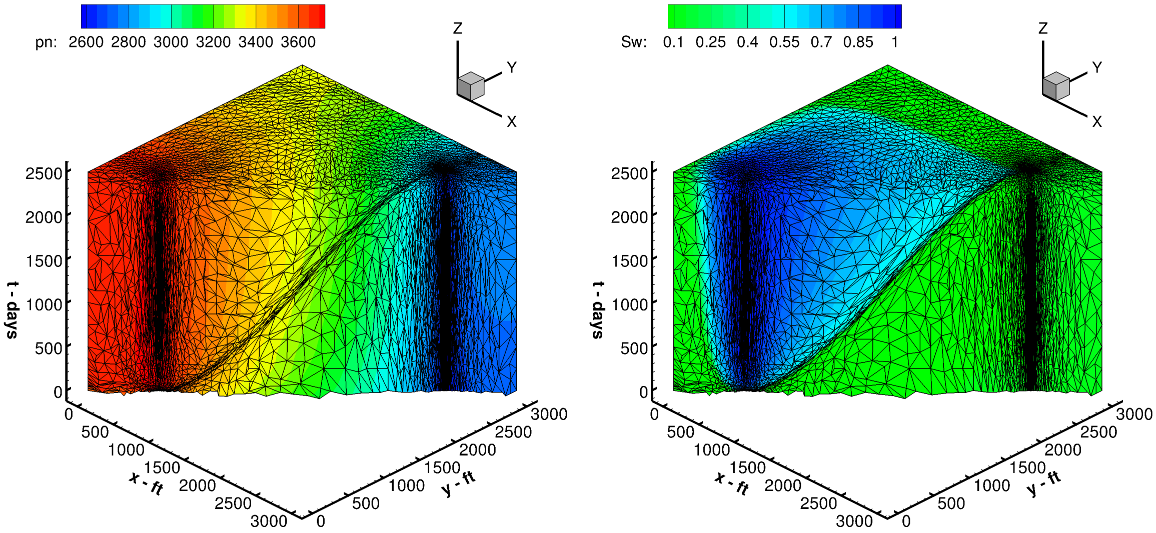
The behavior of the wetting and non-wetting phase adjoint solutions is shown in Figure 6-5, as obtained from the P2 space-time DG adjoint solutions generated for output error estimation. The cone-like structure shown by the blue-green region in Figure 6-5a represents the domain of influence for all adjoint characteristics that propagate backwards in time from the production well. Since the error estimates from the DWR method depend on the approximation errors of both primal and adjoint solutions, the adaptive algorithm focuses on regions where both these errors are large


pn: 2600 2800 3000 3200 3400 3600
Sw: 0.1 0.25 0.4 0.55 0.7 0.85 1
Z
Y
X
t - days
0
500
1000
1500
2000
2500
x - ft
y - ft
3000

(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 6-3: Space-time mesh after 20 iterations of the adaptive algorithm, with P1 DG primal solutions adapted to 10^6 DOF

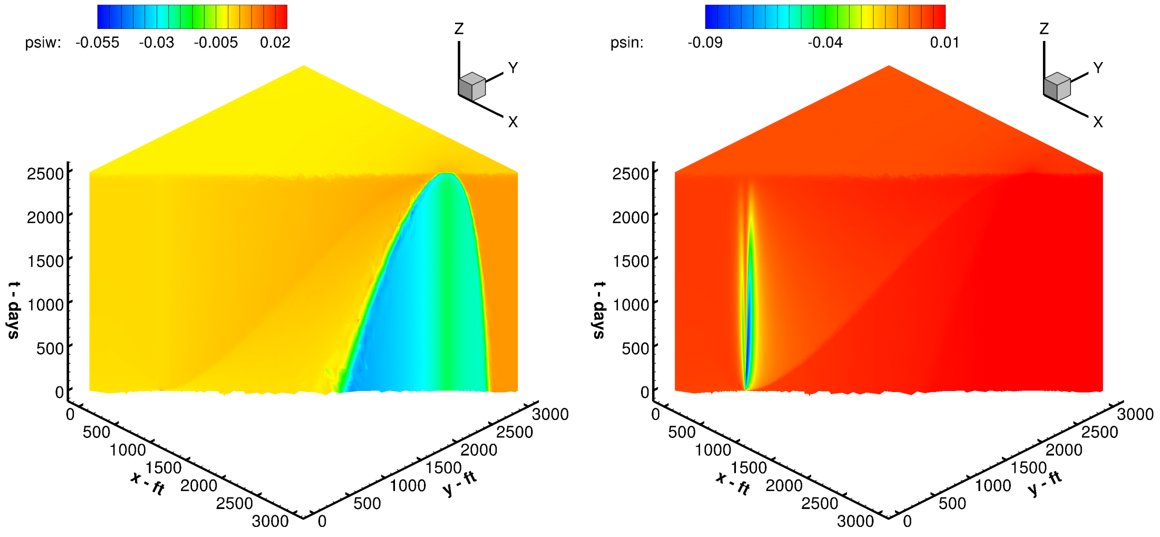

pn: 2600 2800 3000 3200 3400 3600
Sw: 0.1 0.25 0.4 0.55 0.7 0.85 1
Z
Y
X
t - days
0
500
1000
1500
2000
2500
x - ft
y - ft
3000

(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 6-4: Cross-section of final adapted space-time mesh, with P1 DG primal solutions adapted to 10^6 DOF

(e.g. around the wells and along the water saturation front). Figures 6-6 to 6-8 show cross-sections of the P1 adapted space-time DG solution at $t = 625, 1250$ and 1875 days, respectively. The crinkled slices of the space-time mesh at different times, and the corresponding pressure and saturation contours show how the spatial resolution of the mesh evolves with the solution. Although the nearly-steady pressure behavior causes the resolution around the wells to remain fixed throughout the simulation, the mesh resolution around the saturation front closely tracks the front as it moves across the spatial domain.

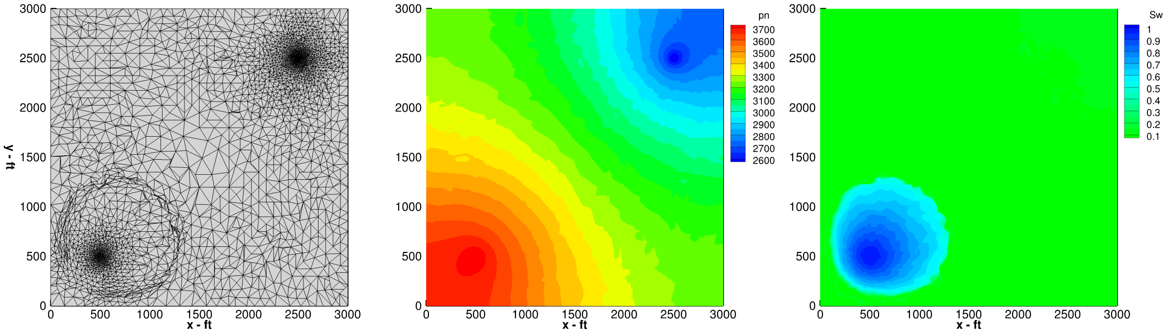


(a) Wetting phase adjoint ψ_w

(b) Non-wetting phase adjoint ψ_n

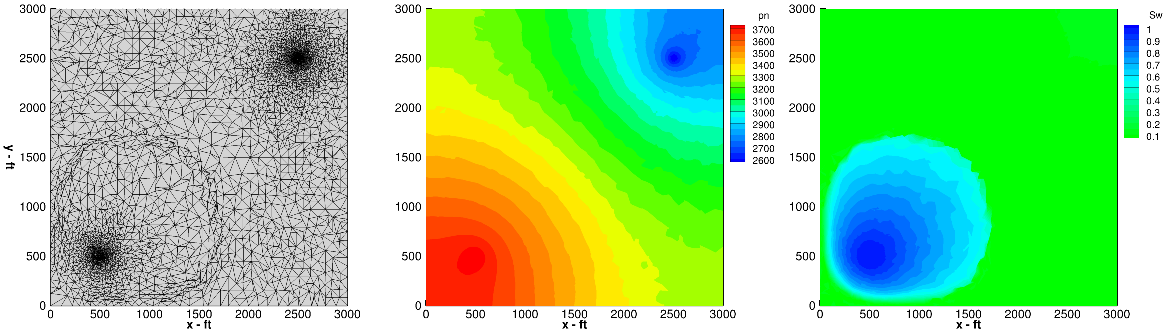
Figure 6-5: Cross-sections of P2 DG adjoint solutions in the space-time domain

Figure 6-9 shows the convergence of the oil recovery factors predicted by each of the seven discretizations with mesh refinement. For the adaptive methods, the mean oil recovery factor of the last five adapted meshes is plotted. The number next to a data point on the plot shows the total number of space-time DOFs in that discrete solution. In terms of total space-time degrees of freedom required to achieve an error tolerance of $\pm 0.1\%$ (marked by the dotted horizontal lines), the time-marching finite volume method (black dashed line) requires $\sim 10^9$ space-time DOF, whereas the P2 adapted space-time DG method (solid green line) requires only $\sim 10^5$ unknowns. Figure 6-10 shows the error convergence behavior of each discretization, where the output error $\mathcal{E}$ is defined relative to the output J from a reference solution computed using an adapted P2 space-time DG solution containing 1 million DOF. As before, the non-dimensionalized average mesh size h is used to fairly compare the time-marching methods with the space-time methods. The time-marching finite volume method is observed to have a first order convergence rate, which is likely dominated by the temporal errors of the backward Euler scheme. All of the P1 DG methods exhibit



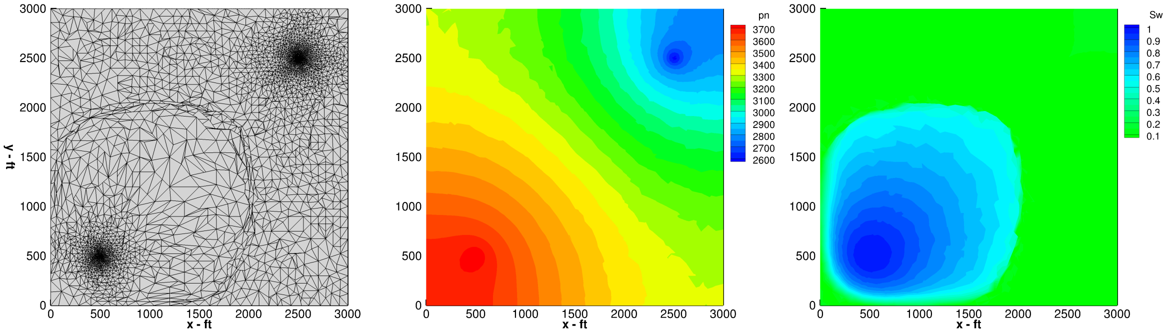
(a) Mesh cross-section (b) Pressure p_n (c) Saturation S_w

Figure 6-6: Slice of P1 space-time DG solution at $t = 625$ days



(a) Mesh cross-section (b) Pressure p_n (c) Saturation S_w

Figure 6-7: Slice of P1 space-time DG solution at $t = 1250$ days



(a) Mesh cross-section (b) Pressure p_n (c) Saturation S_w

Figure 6-8: Slice of P1 space-time DG solution at $t = 1875$ days

second-order convergence rates as expected, with the P1 adapted space-time DG method being the most efficient among them. Comparing the space-time DG results between structured and adapted meshes shows that mesh adaptation enables the DG method to achieve super-convergence in the output with significantly lower errors for the same h. Overall, the P2 adapted space-time DG method (solid green line) is the most efficient in terms of degrees of freedom required to achieve a given level of error.

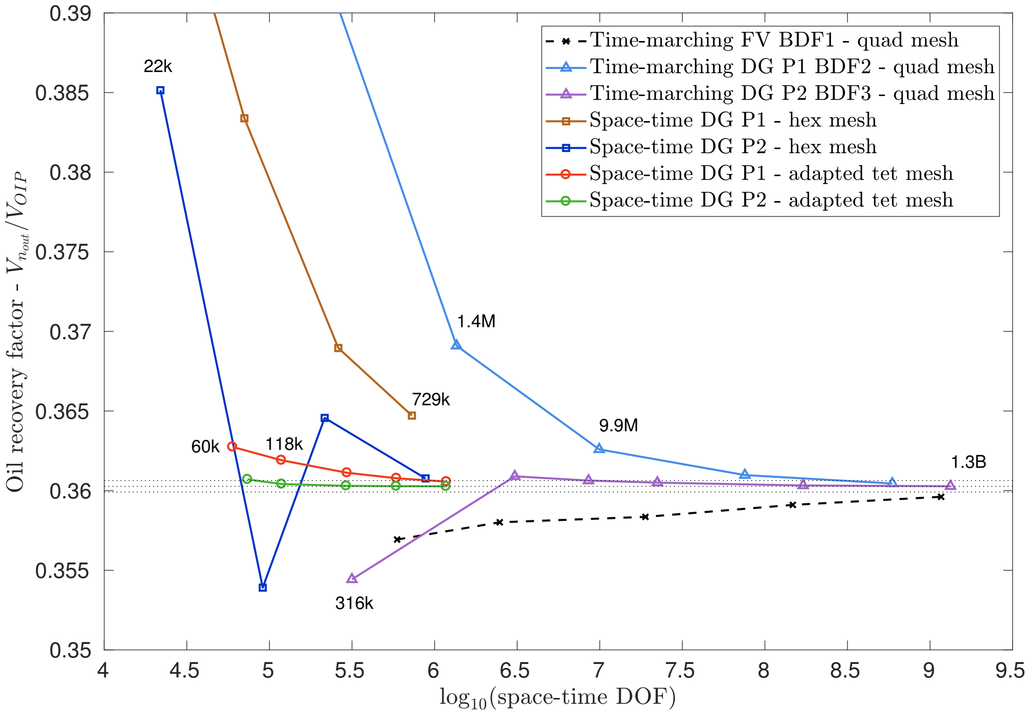


Figure 6-9: Output vs. space-time DOF for different discretizations

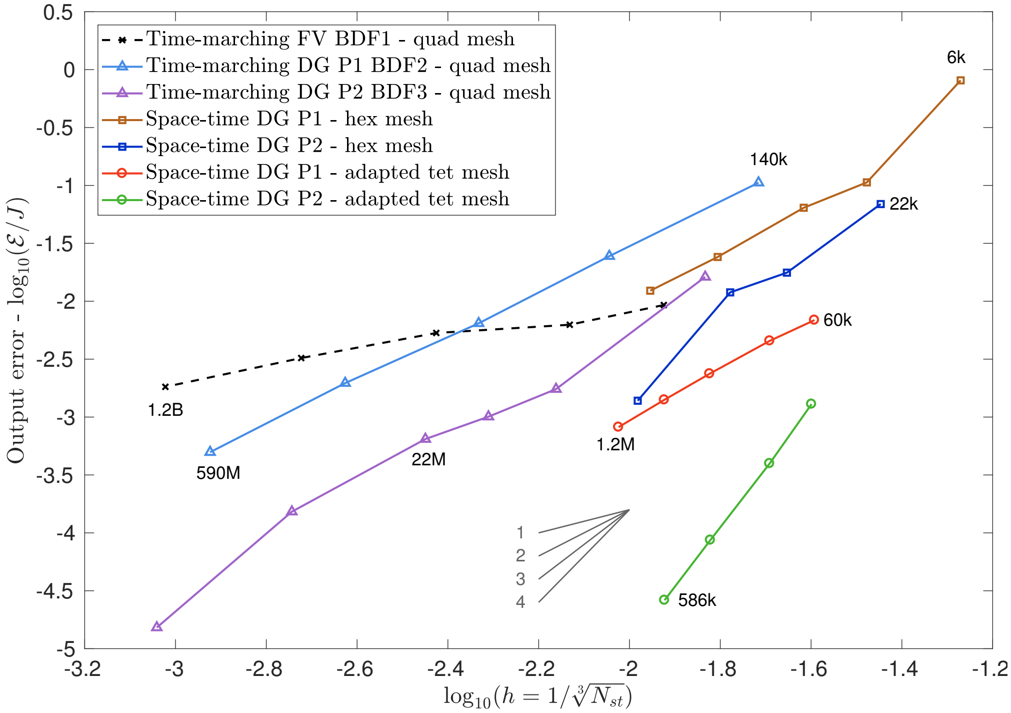


Figure 6-10: Output error vs. mesh size h for discretizations

The relative performances of the different discretizations are analyzed further by comparing the wall-clock times taken to achieve a 0.1% error level. For each line in Figure 6-10, the case with an error level closest to 0.1% is selected, and executed with 2 to 64 parallel MPI processes. All cases were run on a high performance computing (HPC) cluster where each compute node contained two Intel Xeon E5-2660 v3 (2.6 GHz, 10-core) processors, each with a 25 MB cache. The wall-clock times of each of these simulations are shown in Figure 6-11. The timing data shown for the adaptive space-time DG method represent the total time taken for 20 intermediate mesh adaptation iterations, which include the time taken for the primal solves, adjoint solves, error estimation, and local sampling. In contrast, the data of all the remaining methods only represent the time taken for a single primal solve.

The time-marching FV method, the P2 space-time DG method on structured hex meshes, and the P1 adaptive space-time DG method have comparable wall-clock times, and are also the slowest to achieve a 0.1% error level out of the methods tested. Compared to these methods, the P1 time-marching DG method and the P2 adaptive space-time DG method are faster by factors of roughly 5 and 10 respectively. The P2 time-marching DG method is about a further 5 times faster for a small number of processes, but it does not scale well with increasing process count, as seen in Figure 6-12. In general, the space-time DG methods (on both adapted and structured meshes) are observed to have a greater scalability with increasing MPI processes compared to the time-marching methods. This is because the space-time methods can parallelize in both space *and* time, whereas the time-marching methods can only parallelize spatially within a given time-slab. Furthermore, the adaptive space-time DG results also automatically provide additional information, such as adjoint solutions and output error estimates, for which all of the other methods have to do extra work.

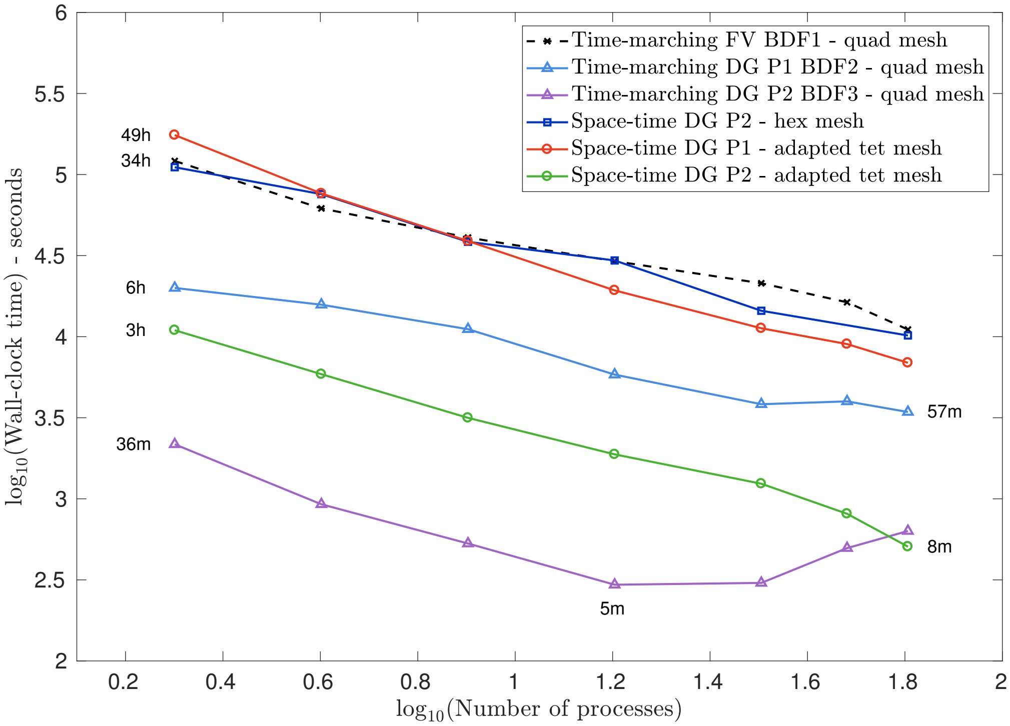


Figure 6-11: Wall time required to achieve 0.1% output error vs. MPI process count

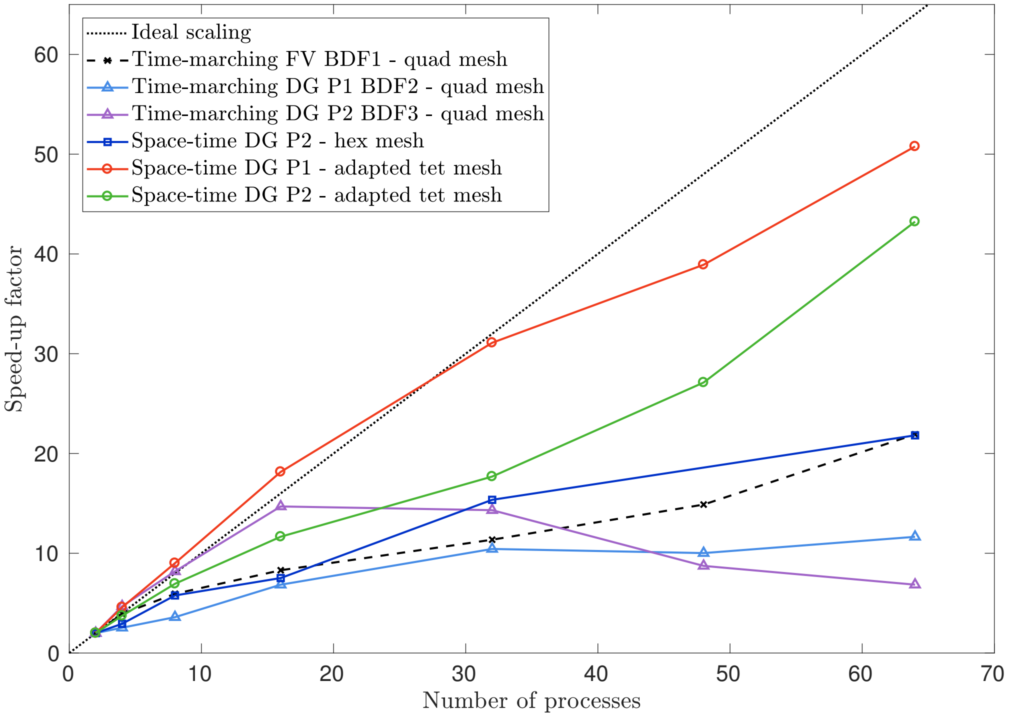


Figure 6-12: Speed-up factor vs. MPI process count

6.3 Summary

In this chapter, the output prediction performance of the adaptive space-time DG method was compared against time-marching and space-time discretizations on structured meshes, for a slightly compressible two-phase flow problem in a homogeneous reservoir. The P2 adaptive space-time DG method, which is third order accurate in both space and time, was found to be the most efficient in terms of degrees of freedom required to achieve a given level of output accuracy. Furthermore, comparisons of wall-clock and CPU times of the different methods showed that the P2 time-marching DG method required the least amount of time to achieve a 0.1% error in the output. However, in terms of parallel scalability, the P1 and P2 adaptive space-time DG methods significantly outperformed the others.

Chapter 7

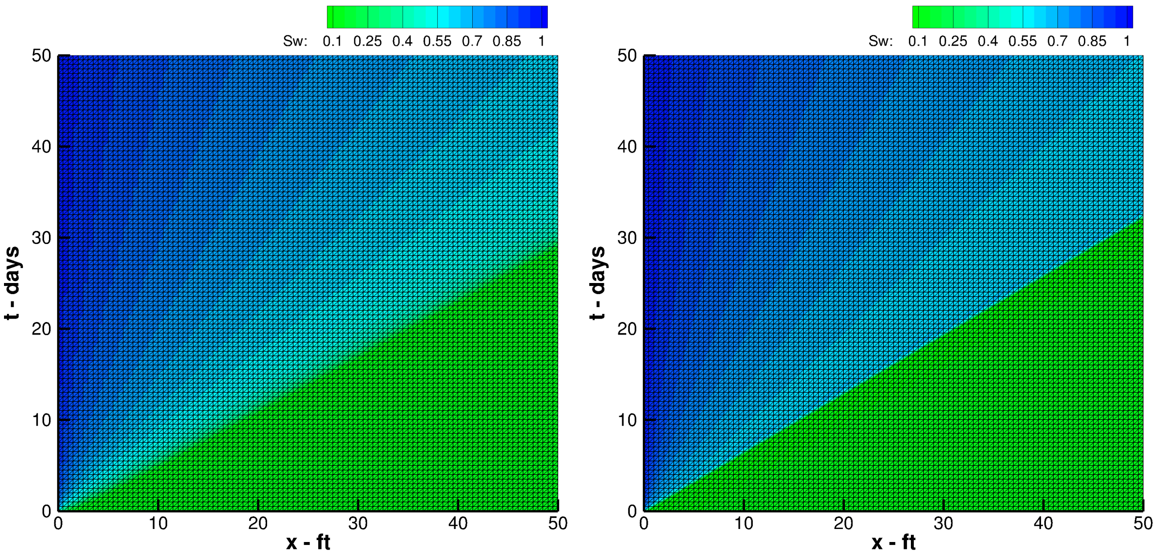
PDE-based artificial viscosity for two-phase flow

High-order discretizations of hyperbolic conservation laws are known to produce spurious oscillations in the vicinity of solution discontinuities, which may also spread to regions of smooth flow and corrupt the accuracy of the solution globally. As discussed in Section 1.3.6, the key ingredient for mitigating these unphysical oscillations is to add dissipation, preferably in a controlled manner that does not affect smooth regions. In this chapter, we extend the PDE-based artificial viscosity method proposed by Barter and Darmofal [17] to high-order space-time DG discretizations of two-phase flow problems.

7.1 Entropy-violating Buckley-Leverett solutions

This section presents space-time DG solutions of the Buckley-Leverett equation, which show that high-order discretizations may *converge* to entropy-violating solutions in the absence of physical or artificial diffusion.

Consider the Buckley-Leverett problem given in Section 3.2, but with a larger final time of $T = 50$ days, such that the saturation front exits the domain out of the right boundary. Figure 7-1 shows the P0 and P1 space-time DG solutions to this problem, on a structured mesh with 20,000 triangle elements. Both solutions show a combined rarefaction-shock wave propagating through the domain, but the saturation front is clearly more diffused in the P0 solution as a result of the greater numerical dissipation. Comparisons of these discrete solutions with the analytic Buckley-Leverett solution are given in Figures 7-2 - 7-3, where the discrete solutions at different times are plotted using solid lines and the analytic solution using dashed lines. The P0 solution, although significantly diffused, appears to predict the correct front speed and has saturation values that are physical. In contrast, the P1 space-time DG method converges to a solution that has an incorrect front speed, an incorrect saturation jump across the front, and significant unphysical oscillations. The flat solution profile downstream of the front is also different to the analytic solution. This residual-converged P1 solution is clearly entropy violating in the sense of Oleinik's entropy condition [91], since it contains characteristics that emanate out of the shock on the downstream side. It should also be noted that the P1 DG method does not converge to an entropy satisfying solution even with further mesh refinement, unless dissipation is added explicitly (e.g. through capillary effects).



(a) P0 solution

(b) P1 solution

Figure 7-1: Space-time DG solutions to the Buckley-Leverett problem, on a structured mesh with 100 × 100 × 2 triangles

Considering that the Buckley-Leverett equation is a simplified model for the saturation transport in the two-phase flow equations, it is reasonable to assume that

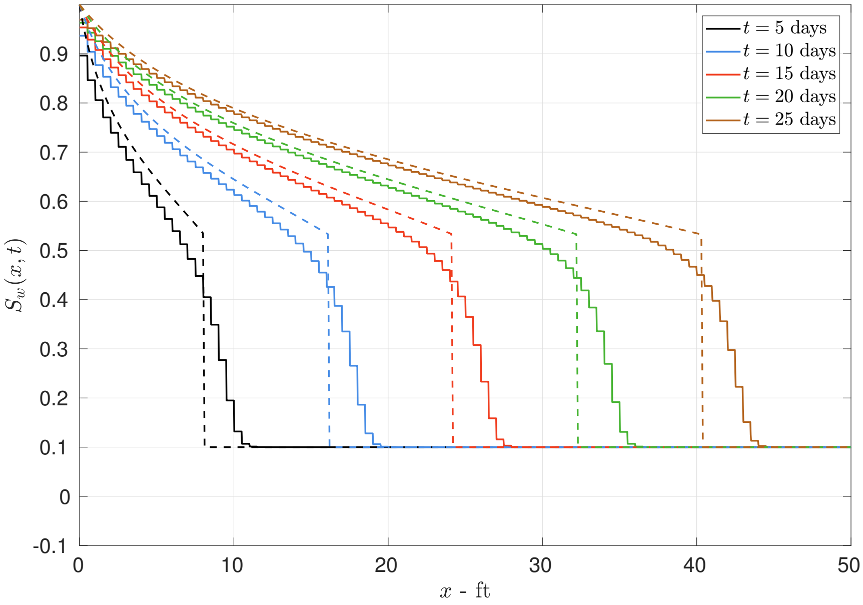


Figure 7-2: Comparison of the P0 space-time DG solution (solid lines) with the analytic solution (dashed lines) at different times

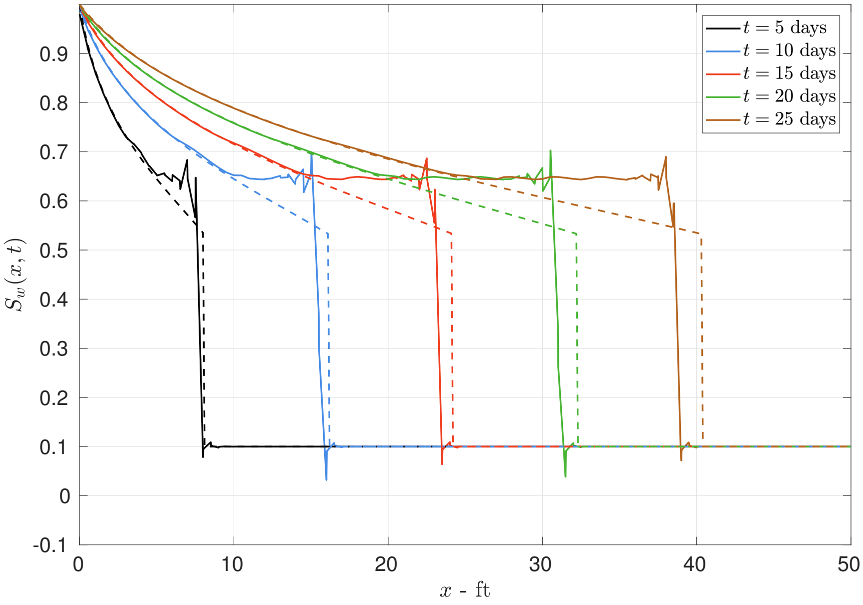


Figure 7-3: Comparison of the P1 space-time DG solution (solid lines) with the analytic solution (dashed lines) at different times

high-order discretizations of the two-phase flow equations may also converge to entropy violating solutions, especially in the limit of vanishing physical diffusion. This therefore motivates the development of a discontinuity-capturing method for the two-phase flow equations.

7.2 Artificial viscosity PDE

This work uses a modified version of the PDE-based artificial viscosity method introduced by Barter and Darmofal [17], where the distribution of artificial viscosity is governed by a reaction-diffusion equation. The reaction term generates artificial viscosity in regions with solution discontinuities, as dictated by a “shock sensor”, and the diffusion term helps to smooth the generated artificial viscosity. This auxiliary PDE is then solved together with the primary equations (e.g. Buckley-Leverett, two-phase flow) in a coupled manner.

In Barter’s original formulation, the artificial viscosity equation is cast as an unsteady equation with a time derivative term, with a time scale chosen such that the artificial viscosity evolves at least as fast as the primary equations. However, most reservoir flow problems, such as the two-phase flows considered in this thesis, are driven by an underlying pressure equation that is elliptic or near-elliptic. Hence, it seems reasonable to modify the artificial viscosity equation to be elliptic as well, thereby eliminating any lag between the need for artificial viscosity and the production of artificial viscosity. Therefore in this work, the hyperbolic temporal term in Barter’s original equation is omitted, and spatial and space-time diffusion approaches for smoothing artificial viscosity are investigated.

7.2.1 Spatial formulation

In this formulation, the artificial viscosity $\nu(\vec{x}, t)$ is governed by the reaction-diffusion PDE given below,

$$-\nabla \cdot \left(C_1 \mathbf{H}^2 \nabla \nu \right) + \nu = f_{\text{switch}}(\xi(s)) \; \nu_{\max}, \tag{7.1}$$

where $\mathbf{H}(\vec{x}) \in \text{Sym}_d^+$ is the generalized length scale tensor that describes the spatial size of the local mesh elements, and C_1 is a constant that roughly determines the number of mesh elements over which artificial viscosity is diffused. In this work, C_1 is set to 3. Although Barter’s original formulation uses axis-aligned bounding boxes to measure the local mesh sizes, the formulation presented here uses Yano's modification to the length scale calculation [137], where the Riemannian metric tensor is used to compute the generalized length scale tensor as $\mathbf{H}(\vec{x}) = \mathcal{M}^{-1/2}(\vec{x})$. The use of tensor-based length scales instead of vector-based length scales has been shown to improve the resolution of shocks on meshes with arbitrarily oriented, highly anisotropic elements [137]. For space-time discretizations, the length scale tensor obtained from the Riemannian metric field contains both spatial and temporal scales of the elements in the space-time mesh, and is hereafter denoted by $\hat{\mathbf{H}}(\hat{\vec{x}}) \in \mathrm{Sym}^{+}_{d+1}$, where $\hat{\vec{x}}$ is the space-time coordinate. For these space-time cases, the spatial length scale tensor required by Eq. 7.1 is obtained by omitting the last row and column of $\hat{\mathbf{H}}$, which contain entries corresponding to the temporal scales.

In this work, the generalized length scale tensor field is represented using a discontinuous, piecewise constant (P0) solution basis, where the length scale tensor value in an element κ is computed as the inverse matrix square root of the elemental metric tensor of κ. This produces a length scale tensor field that is discontinuous between elements for unstructured meshes. Alternatively, the length scale tensor field may also be represented using a continuous, piecewise linear (P1) solution basis, where the $\mathbf{H}$ value at a mesh vertex is computed from a log-Euclidian average of the elemental metric tensors of the elements around that vertex. However, this approach tends to produce more negative artificial viscosity values on unstructured meshes compared to the piecewise constant representation, and is therefore not used in this work.

The production of artificial viscosity is driven by the forcing term on the right-hand side of Eq. 7.1, which becomes non-zero in the vicinity of solution discontinuities. The function $f_{\text{switch}}(\cdot) : \mathbb{R} \to (0, 1)$ is a nonlinear, monotone switch such that $f_{\text{switch}}(0) \approx 0$ and $f_{\text{switch}}(1) \approx 1$. This switch operates on a cell discontinuity sensor s, which quantifies the magnitude of solution discontinuities present around a given mesh element. This cell discontinuity sensor s is then also filtered, via $\xi(s)$, to ignore small discontinuities (i.e. smooth regions) and to cap the discontinuity level at which artificial viscosity production is maximized. In this thesis, s is represented using a scalar, piecewise constant (P0) DG solution, and is computed by solving the following linear local problem on each element κ,

$$\frac{1}{|\kappa|}\int_{\kappa} v\ s\ d\Omega = \sum_{f\in\partial\kappa\cap\Gamma_I}\frac{1}{|f|}\int_{f} v^{+}g(\mathbf{u}^{+},\mathbf{u}^{-})\ d\Gamma, \qquad \forall v\in V_{h,(p=0)}, \tag{7.2}$$

where $|\kappa|$ is the volume of element κ, $|f|$ is the area of interior face f, and $g(\mathbf{u}^{+},\mathbf{u}^{-})$ is a scalar quantity used to detect a shock, which depends on the primary solution variables on either side of a face. Solving Eq. 7.2 on an element κ causes the value of the cell sensor s in that element to be the average value of g over the faces of κ. Since this work is concerned with detecting and stabilizing oscillations in the saturation solution, g is defined to be the absolute value of the jump in saturation across a face,

$$g(\mathbf{u}^{+},\mathbf{u}^{-}) = \left|S_w^{+} - S_w^{-}\right|. \tag{7.3}$$

Solving Eq. 7.2 repeatedly for all elements in the mesh causes interior face integrals to be computed twice. Hence, for computational efficiency, Eq. 7.2 is summed over all elements in the mesh and re-arranged to produce the following alternate weak form,

$$\sum_{\kappa\in\mathcal{T}_h}\int_{\kappa} v\ s\ d\Omega = \sum_{f\in\Gamma_I}\int_{f}\left(\frac{|\kappa^{+}|\,v^{+}}{N_{\text{face}}^{+}\,|f|} + \frac{|\kappa^{-}|\,v^{-}}{N_{\text{face}}^{-}\,|f|}\right) g(\mathbf{u}^{+},\mathbf{u}^{-})\ d\Gamma, \qquad \forall v\in V_{h,(p=0)}, \tag{7.4}$$

where N_{face}^{+} and N_{face}^{-} represent the number of faces of κ^{+} and κ^{-} respectively. Assuming the mass matrix inverse of each element is computed and stored beforehand, this formulation allows the cell sensor s to be computed efficiently through accumulation, using only a single loop over the interior faces of the mesh. Higher-order representations of the cell sensor s can be easily obtained through this formulation by changing the order of the polynomial basis used for v and s, but the effects of doing so are not considered in this thesis.

Once the cell discontinuity sensor s is computed, it is filtered through $\xi(s)$ as follows,

$$\xi(s) = \frac{\log_{10}(|s| + \epsilon) - \log_{10}(s_{\min})}{\log_{10}(s_{\max}) - \log_{10}(s_{\min})}, \tag{7.5}$$

where $s_{\min}$ and $s_{\max}$ represent thresholds for the addition of artificial viscosity. When $\xi(s)$ is fed through the nonlinear activation function $f_{\text{switch}}(\cdot)$, regions with sensor values lower than $s_{\min}$ do not produce any artificial viscosity, whereas regions with sensor values greater than $s_{\max}$ produce the maximum amount of artificial viscosity. In this work, $s_{\min} = 10^{-3}$, $s_{\max} = 10^{-1}$ and $\epsilon = 10^{-16}$.

The last component of the artificial viscosity PDE is $\nu_{\max}$, which represents the maximum amount of artificial viscosity that is added to the primary equations. As proposed by Persson and Peraire in [118], $\nu_{\max}$ needs to be $\mathcal{O}(h/p)$ to allow for sub-cell shock resolution, where h is the local mesh size and p is the polynomial order of the solution. The form of $\nu_{\max}$ used in this work is based on a multi-dimensional and higher-order extension of the grid Péclet number, given by,

$$\nu_{\max}(\mathbf{u}) = \frac{\sqrt{\vec{v}^T \mathbf{H}^2 \vec{v}}}{2p}, \tag{7.6}$$

where $\mathbf{H}(\vec{x})$ is the generalized spatial length scale tensor from before, $\vec{v}(\mathbf{u})$ is a problem-dependent characteristic velocity, and p is the solution order. The maximum artificial viscosity is designed to achieve a local grid Péclet number of Pe $= 2$, inspired from the stability theory of second-order finite difference schemes for 1D linear advection-diffusion problems.

The boundary conditions of the artificial viscosity PDE were specified to be of the radiation-type (i.e. Robin) by Barter and Darmofal [17], where the flux of ν is proportional to the difference between the boundary value and an ambient viscosity value (ν_∞), multiplied by a local length scale. Since it is possible for a shock or discontinuity to leave the domain at an arbitrary angle to the boundary, homogeneous Dirichlet ($\nu = 0$) or Neumann ($\frac{\partial \nu}{\partial \vec{n}} = 0$) BCs are not appropriate. The strong form of the Robin BC used in this work is given by,

$$-\left(C_1\mathbf{H}^2\nabla\nu\right)\cdot\vec{n} = \sqrt{C_1}\left(\vec{n}^T\mathbf{H}\vec{n}\right)(\nu-\nu_\infty), \tag{7.7}$$

where $\vec{n}$ is the outward unit normal vector of the spatial boundary, and the ambient viscosity value is set to zero ($\nu_\infty = 0$). Since the term on the left-hand side of Eq. 7.7 is exactly the diffusive flux of the artificial viscosity PDE, this BC is discretized by setting the discrete boundary flux equal to the term on the right-hand side, where ν is evaluated from the interior of the domain ($\nu^B = \nu^+$).

7.2.2 Space-time formulation

For space-time discretizations, it is possible to modify the "spatial" artificial viscosity formulation described in Section 7.2.1 to include diffusive effects in the temporal direction as well. Although this seems counter-intuitive, this idea arises naturally if the primary equations are thought of as "steady" state equations in a $(d+1)$-dimensional domain, instead of as unsteady equations in a d-dimensional domain. For example, a discontinuity that is traveling across a 1D spatial domain at a constant speed can be viewed as a stationary discontinuity on a 2D space-time domain that lies at some angle to the spatial axis (e.g. the saturation front in Figure 7-1b). The idea behind a "space-time" artificial viscosity formulation is that such a discontinuity can be diffused or smeared-out in a direction that is orthogonal to the discontinuity in the space-time domain, as opposed to diffusing it only spatially.

In order to achieve this, the artificial viscosity PDE presented in Section 7.2.1 is modified as follows,

$$-\hat{\nabla}\cdot\left(C_1\hat{\mathbf{H}}^2\hat{\nabla}\nu\right) + \nu = f_{\text{switch}}(\xi(s))\ \nu_{\max}, \tag{7.8}$$

where the only difference is that all spatial gradient operators, $\nabla(\cdot)$, are replaced with space-time gradient operators, $\hat{\nabla}(\cdot)$, and the spatial length scale tensor $\mathbf{H}$ is replaced with its space-time counterpart $\hat{\mathbf{H}}$, which is obtained from the Riemannian metric field of the space-time mesh. As before, $\hat{\mathbf{H}}(\vec{x})$ is represented using a discontinuous, piecewise constant (P0) solution basis on each space-time element. The definition of the maximum artificial viscosity in Eq. 7.6 is left unchanged, and still uses the spatial length scale tensor $\mathbf{H}$. Eq. 7.8 is elliptic in space-time, and therefore diffuses the artificial viscosity produced by the source term both spatially and temporally.

The Robin boundary condition given previously in Eq. 7.7 is also modified to the following,

$$-\left(C_1 \hat{\mathbf{H}}^2 \hat{\nabla} \nu\right) \cdot \hat{\vec{n}} = \sqrt{C_1} \left(\hat{\vec{n}}^T \hat{\mathbf{H}} \hat{\vec{n}}\right) (\nu - \nu_\infty), \tag{7.9}$$

where $\hat{\vec{n}}$ represents the unit space-time normal vector of the boundary. Numerical results from both "spatial" and "space-time" artificial viscosity formulations are presented and compared in the next section.

7.3 Artificial viscosity for Buckley-Leverett

This section presents results for the Buckley-Leverett problem considered previously in Section 7.1, but now modified with the proposed artificial viscosity term to help mitigate unphysical oscillations that may occur when high-order discretizations are used. The original Buckley-Leverett equation is modified as follows,

$$\frac{\partial}{\partial t}\left(\phi S_w\right) + \frac{\partial}{\partial x}\left(u_T f_w(S_w)\right) = \frac{\partial}{\partial x}\left(\phi \tilde{\nu} \frac{\partial S_w}{\partial x}\right), \tag{7.10}$$

where,

$$\tilde{\nu}(\nu) = \text{smoothmax}(\nu, 0), \tag{7.11}$$

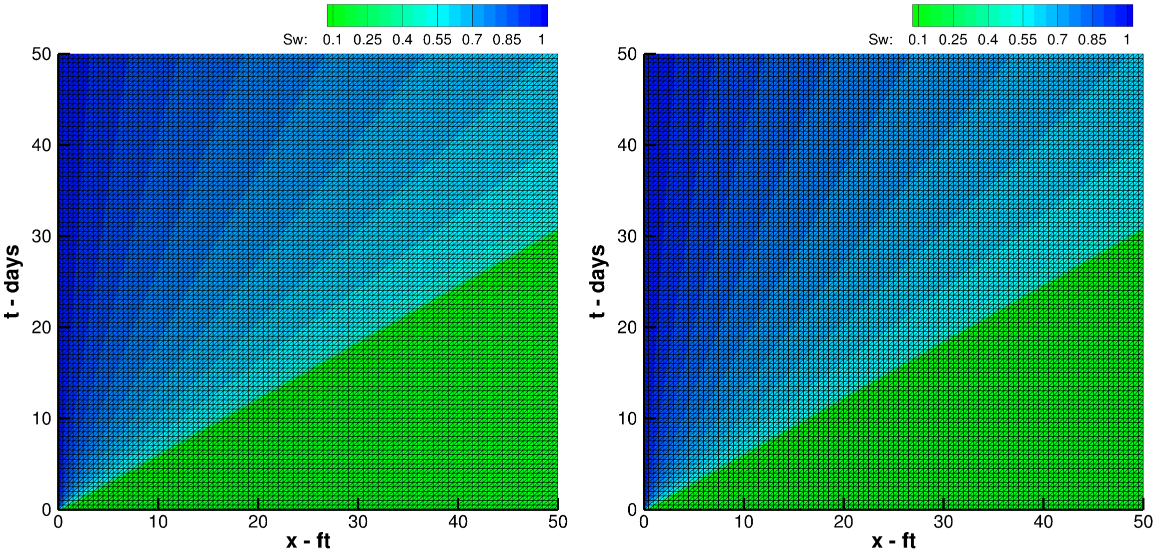
and the artificial viscosity $\nu(x,t)$ is an auxiliary variable that is governed by the artificial viscosity PDE. The primary state vector is augmented to include the artificial viscosity (i.e. $\mathbf{u} = [S_w, \nu]^T$) and Eq. 7.10 is solved in a coupled manner with one of the artificial viscosity PDEs presented in the previous subsection. The artificial viscosity variable is passed through a smoothmax function, as given by Eq. 7.11, to ensure that the viscosity added to the Buckley-Leverett equation is always positive, since it is possible for a piecewise polynomial representation of ν to produce negative values, especially on highly anisotropic elements. The characteristic velocity of the Buckley-Leverett equation required for the calculation of $\nu_{\max}$ is given by,

$$\vec{v} = \frac{u_T}{\phi} \frac{df_w}{dS_w}, \tag{7.12}$$

where $f_w(S_w)$ is the fractional flow function as given in Section 3.2. The coupled system of equations is then solved for $\mathbf{u}(x,t)$ using a piecewise linear (P1) space-time DG scheme on a structured triangle mesh as before.

Figures 7-4 to 7-7 offer side-by-side comparisons of the solutions obtained from the "spatial" and "space-time" artificial viscosity formulations. The wetting phase saturation contours given in Figure 7-4 show the combined rarefaction-shock wave propagating through the domain, with sharp resolution of the saturation front. The artificial viscosity contours in Figure 7-5 show that the spatial formulation produces a larger artificial viscosity at the discontinuity compared to the space-time formulation. This also implies that the saturation front is slightly more diffused for the spatial formulation than for the space-time formulation. Figure 7-6 shows zoomed-in views of the artificial viscosity contours at the right boundary where the saturation front exits the domain. The presence of solely a spatial diffusion operator in the spatial formulation causes the artificial viscosity to be diffused spatially within a given time-slab, and be decoupled from other time-slabs, as seen in Figure 7-6a. As a result, the distribution of ν is forced to be orthogonal to the boundary, which is inconsistent with the exit angle of the discontinuity. Using the space-time formulation diffuses the artificial viscosity in a direction that is orthogonal to the discontinuity, as seen by the contours in Figure 7-6b. The space-time boundary condition also appears to be more consistent with the exit angle of the saturation front. Figure 7-7 shows the distribution of the piecewise constant cell sensor s, in the log scale, where large s values illuminate the path of the saturation front in both formulations. Figures 7-8 and 7-9 contain cross-sections of the space-time solutions of S_w and ν at different times, using the spatial and space-time artificial viscosity formulations respectively. As before, the dashed line on each plot shows the analytic Buckley-Leverett solution. Comparing these saturation solutions with Figure 7-3 shows that the addition of artificial viscosity enables the second-order P1 DG scheme to converge a physical solution that has less dissipation (and therefore sharper fronts) compared to the P0 result. The artificial viscosity solutions plotted at the bottom of Figures 7-8 and 7-9 further show that the space-time formulation produces a smaller peak value of ν at the front compared to the spatial formulation, which is diffused over similar length scales.

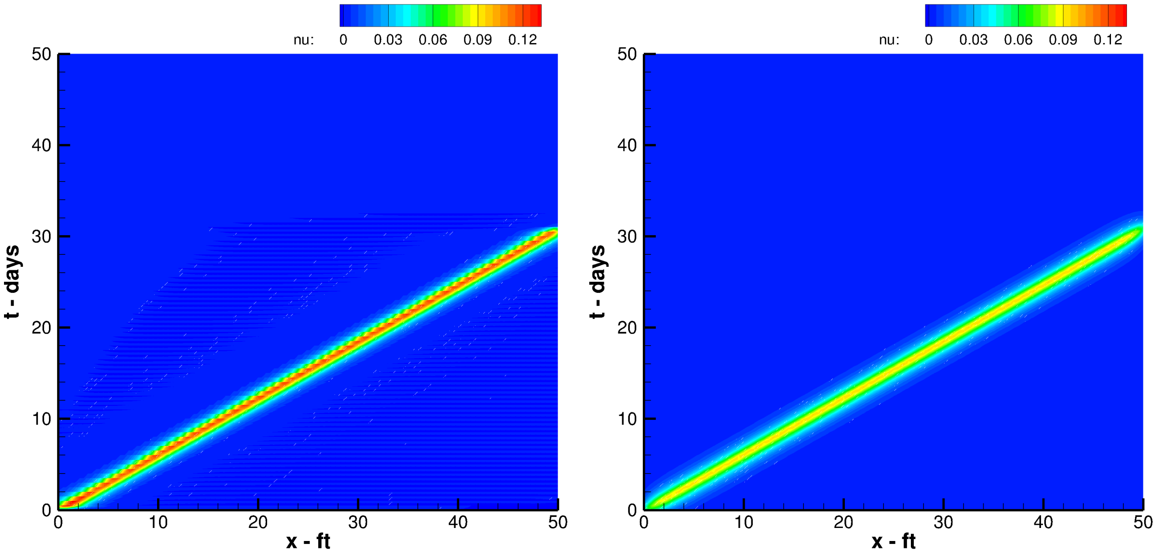
It was also observed that the space-time formulation typically requires fewer nonlinear solver (i.e. Newton) iterations to converge, and is generally more robust compared to the spatial artificial viscosity formulation. Hence, for the remainder of this thesis, the space-time artificial viscosity formulation described in Section 7.2.2 is used.



(a) Spatial artificial viscosity

(b) Space-time artificial viscosity

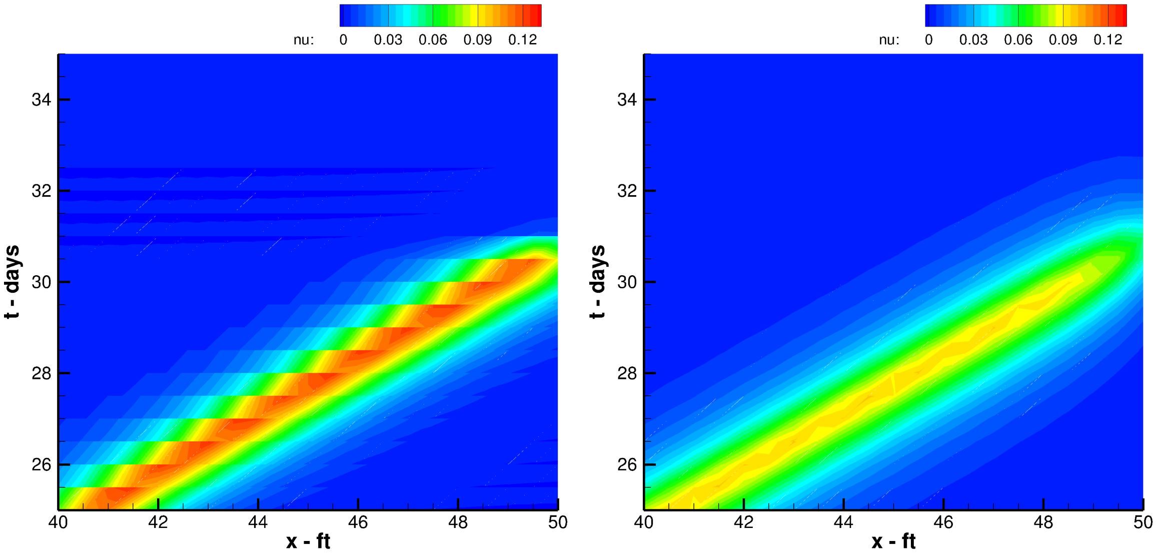
Figure 7-4: Saturation contours from P1 space-time DG solutions on a structured mesh with 100 × 100 × 2 triangles, using different artificial viscosity formulations



(a) Spatial artificial viscosity

(b) Space-time artificial viscosity

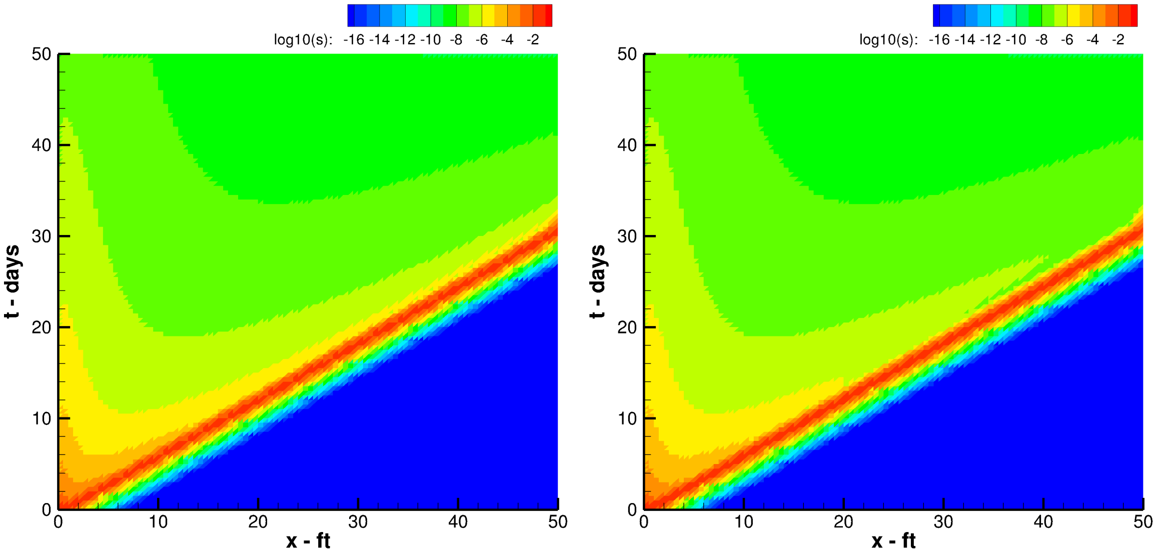
Figure 7-5: Artificial viscosity contours from P1 space-time DG solutions on a structured mesh with $100 \times 100 \times 2$ triangles, using different artificial viscosity formulations



(a) Spatial artificial viscosity

(b) Space-time artificial viscosity

Figure 7-6: Zoomed-in views of artificial viscosity contours from P1 space-time DG solutions on a structured mesh with $100 \times 100 \times 2$ triangles, using different artificial viscosity formulations



(a) Spatial artificial viscosity

(b) Space-time artificial viscosity

Figure 7-7: Piecewise constant cell sensor contours in the log scale, obtained using different artificial viscosity formulations

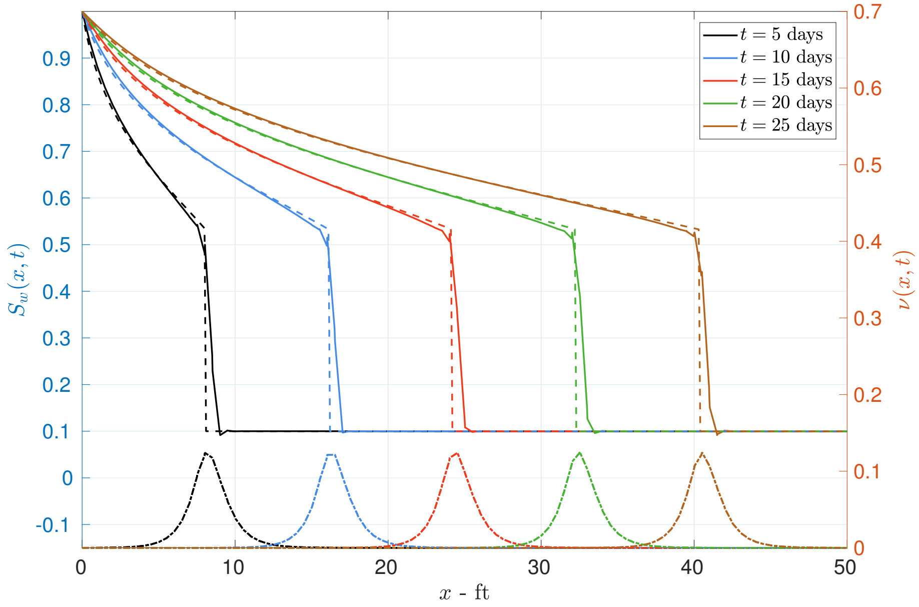


Figure 7-8: Comparison of the P1 space-time DG S_w solution (solid lines) with the analytic solution (dashed lines) at different times, with the ν solution plotted at the bottom. These solutions were obtained using the spatial artificial viscosity formulation.

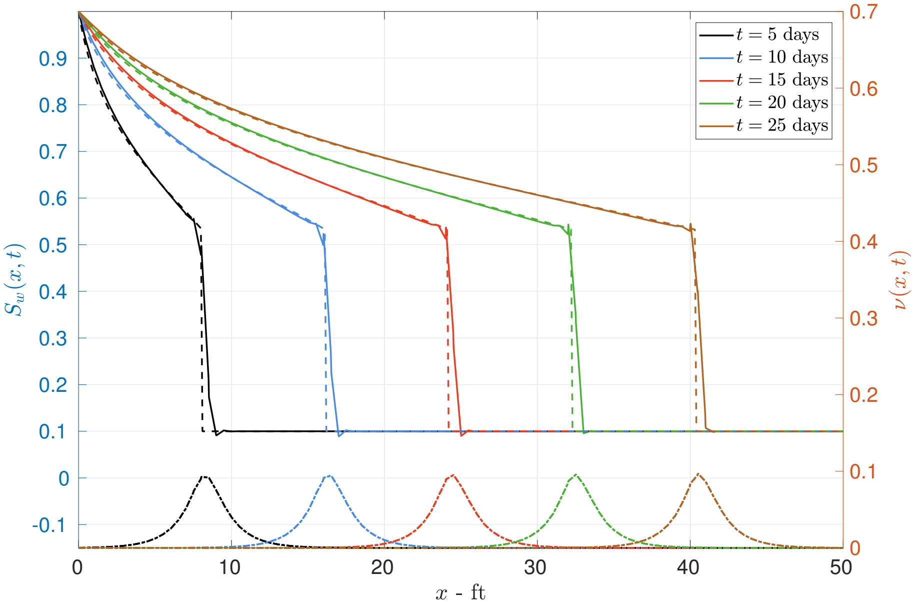


Figure 7-9: Comparison of the P1 space-time DG S_w solution (solid lines) with the analytic solution (dashed lines) at different times, with the ν solution plotted at the bottom. These solutions were obtained using the space-time artificial viscosity formulation.

7.4 Artificial viscosity for two-phase flow

This section describes a method for adding artificial viscosity to the two-phase flow equations in mass conservation form, in order to mitigate Gibbs oscillations in the saturation solution. Since the hyperbolic nature of the two-phase flow equations originates from the underlying saturation transport equation, it is important to add artificial viscosity in a manner that stabilizes the saturation behavior, while minimizing its effect on the elliptic pressure behavior for accuracy.

Intuitively, a capillary pressure based (i.e. physics-based) artificial viscosity model seems a reasonable choice, where any existing capillary pressure is augmented with an artificial viscosity of the same form, i.e. by substituting $(p_{c_S} - \nu)$ for p_{c_S} in Eq. 4.1. Linearizing the two-phase flow equations with this form of artificial viscosity, as done previously in Section 4.1, shows that the diffusion coefficient of the linearized saturation equation is,

$$\bar{\mathbf{D}} = \frac{\bar{\lambda}_w \bar{\lambda}_n}{\phi(\bar{\lambda}_w + \bar{\lambda}_n)}(\nu - \bar{p}_{c_S})\mathbf{K}, \tag{7.13}$$

where all quantites with an overbar $\bar{(\cdot)}$ are evaluated at the mean state. Note that $\bar{p}_{c_S}$ is non-positive, so the diffusion coefficient $\bar{D}$ is non-negative as long as the artificial viscosity ν is non-negative. However, the presence of a product of relative mobilities in the numerator of Eq. 7.13 implies that $\bar{\mathbf{D}}$ vanishes in the limit of $\bar{S}_w \to 0$ or $\bar{S}_w \to 1$, regardless of the values of ν and $\bar{p}_{c_S}$. Figure 7-10 shows examples of how the mobility-dependent term in Eq. 7.13 varies with S_w, for different phase viscosity ratios. As a result, this form of artificial viscosity is unable to add dissipation and stabilize regions with saturation values close to zero or one, which unfortunately, are also the regions that are most likely to give rise to unphysical saturations.

Therefore in this work, we propose a slightly more contrived artificial viscosity formulation for the two-phase flow equations, which is capable of adding dissipation in a more uniform manner for all values of saturation. Consider the following form

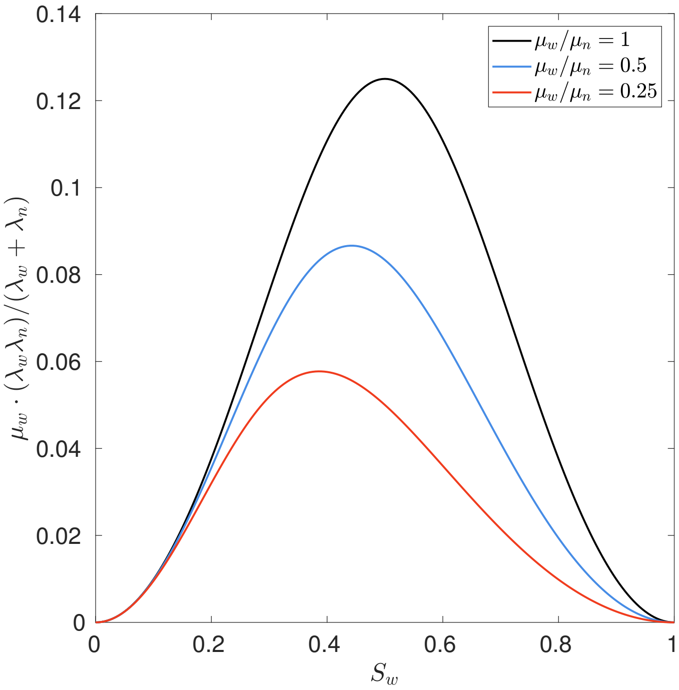


Figure 7-10: Plot of $\mu_w \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n}$ vs. S_w using quadratic relative permeability functions, for different phase viscosity ratios

of the two-phase flow equations modified with artificial viscosity,

$$\frac{\partial}{\partial t}\left(\rho_w \phi S_w\right) - \nabla \cdot \left(\rho_w \lambda_w \mathbf{K} \nabla p_w + \rho_w \nu \nabla S_w\right) = 0, \tag{7.14}$$

$$\frac{\partial}{\partial t}\left(\rho_n \phi (1 - S_w)\right) - \nabla \cdot \left(\rho_n \lambda_n \mathbf{K} \nabla p_n - \rho_n \nu \nabla S_w\right) = 0, \tag{7.15}$$

where the artificial viscosity $\nu \geq 0$. Furthermore, unlike for the capillary pressure-based artificial viscosity formulation described earlier, taking the weighted sum of the augmented equations, $\rho_n \times$ (Eq. 7.14) + $\rho_w \times$ (Eq. 7.15), in the incompressible limit causes the artificial viscosity terms to cancel each other out. This shows that the addition of this form of artifical viscosity does not affect the underlying pressure equation.

In order to complete the description of the artificial viscosity formulation, a definition of the maximum artificial viscosity $\nu_{\max}$ is also required. In this work, $\nu_{\max}$ is defined such that when $\nu = \nu_{\max}$ and capillary effects are ignored (i.e. $p_c = 0$), the linearized saturation equation has a grid Péclet number equal to $\mathrm{Pe} = 2$. Using the multi-dimensional and high-order extension of the grid Péclet number for the linearized saturation equation yields the following expression for $\nu_{\max}$,

$$\frac{\nu_{\max}}{\phi} = \frac{\sqrt{\vec{\bar{V}}^T \mathbf{H}^2 \vec{\bar{V}}}}{2p}, \tag{7.16}$$

where $\vec{\bar{V}}$ is the saturation advection velocity given in Eq. 4.10. However, the presence of relative mobilities $(\bar{\lambda}_w, \bar{\lambda}_n)$ and their derivatives $(\bar{\lambda}_{w_S}, \bar{\lambda}_{n_S})$ in the expression of $\vec{\bar{V}}$ makes the expression for $\nu_{\max}$, and consequently also the artificial viscosity PDE, highly nonlinear with respect to saturation. This nonlinearity is alleviated for better discretization robustness, at the expense of using a more conservative velocity estimate, which does not contain the nonlinear relative permeability functions. Hence, the saturation advection velocity $\vec{\bar{V}}$ is replaced with $\vec{V}_{\max}$ given below,

$$\vec{V}_{\max} = \arg \max_{\vec{V}_\alpha \in \{\vec{V}_w, \vec{V}_n\}} \left(\left| \vec{V}_\alpha \right| \right), \tag{7.17}$$

where $\vec{V}_\alpha$ represents the seepage velocity of phase α,

$$\vec{V}_\alpha = -\frac{1}{\phi} \frac{\mathbf{K} \nabla p_\alpha}{\mu_\alpha}. \tag{7.18}$$

The physical significance of $\vec{V}_{\max}$ above is that it represents the seepage velocity of the faster moving phase. For a problem with no capillary effects (i.e. $p_w = p_n$), $\vec{V}_{\max}$ is always the seepage velocity of the less viscous phase.

The final form of the two-phase flow equations augmented with the space-time artificial viscosity PDE is given below,

$$\begin{aligned} \frac{\partial}{\partial t}\left(\rho_w \phi S_w\right) - \nabla \cdot \left(\rho_w \lambda_w \mathbf{K} \nabla p_w + \rho_w \tilde{\nu} \nabla S_w\right) &= 0, \\ \frac{\partial}{\partial t}\left(\rho_n \phi (1 - S_w)\right) - \nabla \cdot \left(\rho_n \lambda_n \mathbf{K} \nabla p_n - \rho_n \tilde{\nu} \nabla S_w\right) &= 0, \\ -\hat{\nabla} \cdot \left(C_1 \hat{\mathbf{H}}^2 \hat{\nabla} \nu\right) + \nu &= f_{\text{switch}}(\xi(s))\ \nu_{\max}, \end{aligned} \tag{7.19}$$

where $\mathbf{u} = [p_n, S_w, \nu]^T$ is the augmented state vector, and $\tilde{\nu}$ represents the value of viscosity obtained after passing ν through the smoothmax function given in Eq. 7.11. As before, $\tilde{\nu}$ ensures that the artifical dissipation added to the primary equations is positive. Numerical results obtained using this formulation are presented in the next chapter.

Chapter 8

Compressible two-phase flow in a heterogeneous reservoir

This chapter presents numerical results for a slightly compressible two-phase flow problem in a two-dimensional heterogeneous reservoir. This problem is challenging for high-order discretizations due to the presence of a discontinuous rock permeability field and the absence of physical diffusion (zero capillary effects). In contrast to the homogeneous problem considered in Chapter 6, the problem considered here requires artificial viscosity to robustly converge high-order discretizations. Output predictions from the adaptive space-time DG method with artificial viscosity are compared to those from conventional low-order time-marching methods (without artificial viscosity) for computational efficiency.

8.1 Problem statement

The problem considered in this chapter involves a slightly compressible water-oil system that is driven by a two-spot well configuration in a square, heterogeneous reservoir. The spatial domain is given by $\Omega_s = [0, L] \times [0, L]$, where $L = 3000$ ft. The schematic in Figure 8-1 shows the locations of the injection and production wells, at coordinates $(500, 500)$ ft and $(2500, 2500)$ ft, with fixed bottom-hole pressures of 4000 psi and 2000 psi respectively. The darker region in the figure represents the quadrilateral heterogeneity in the absolute rock permeability field, where the permeability is 100 times smaller compared to the rest of the reservoir, and thus manifests as a barrier to the flow between the wells. The corners of the quadrilateral block are at spatial coordinates $(800, 1400)$, $(2000, 1000)$, $(2300, 1300)$ and $(1100, 2200)$ ft. The permeability changes discontinuously across the boundary of the quadrilateral, representing an abrupt change in the geology.

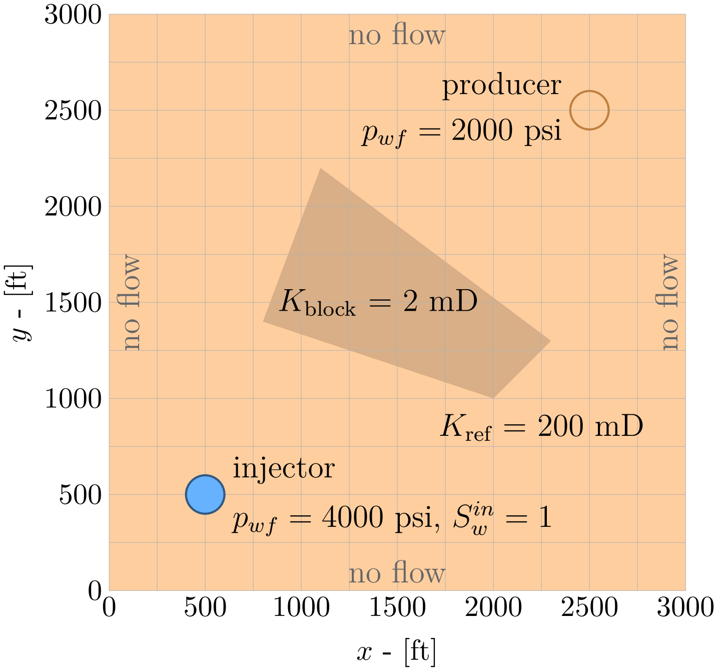


Figure 8-1: Schematic of heterogeneous 2D reservoir

The governing two-phase flow equations, no flow boundary conditions and initial conditions are given by Eqs. 5.55 - 5.58, with the relevant constitutive relationships and numerical constants are replicated below,

$$\rho_\alpha = \rho_{\alpha_{\text{ref}}}\ e^{c_\alpha (p_\alpha - p_{\text{ref}})} \qquad \text{for } \alpha \in \{w, n\}, \tag{8.1}$$

$$\phi = \phi_{\text{ref}}\ e^{c_\phi (p_n - p_{\text{ref}})}, \tag{8.2}$$

$$k_{r\alpha} = S_\alpha^2, \qquad \text{for } \alpha \in \{w, n\} \tag{8.3}$$

$$p_c = p_{c_{\max}} (1 - S_w), \tag{8.4}$$

where,

$$\begin{aligned}
\phi_{\text{ref}} &= 0.3, & p_{\text{ref}} &= 14.7 \text{ psi},\\
\rho_{w_{\text{ref}}} &= 62.4 \text{ lb/ft}^3, & \rho_{n_{\text{ref}}} &= 52.1 \text{ lb/ft}^3,\\
c_w &= 5 \times 10^{-6} \text{ psi}^{-1}, & c_n &= 1.5 \times 10^{-5} \text{ psi}^{-1},\\
c_\phi &= 3 \times 10^{-6} \text{ psi}^{-1}, & p_{c_{max}} &= 5 \text{ psi},\\
\mathbf{K}_{\text{ref}} &= 200\ \mathbf{I} \text{ mD}, & \mathbf{K}_{\text{block}} &= 2\ \mathbf{I} \text{ mD},\\
\mu_w &= 1 \text{ cP}, & \mu_n &= 2 \text{ cP},\\
r_w &= 2 \text{ inches.}
\end{aligned}$$

As before, the objective of this problem is to accurately predict the oil recovery factor given by,

$$J = \frac{V_{n_{out}}}{V_{OIP}}, \tag{8.5}$$

where $V_{n_{out}}$ is the total volume of oil extracted from the production well over a period of $T = 4000$ days, and V_{OIP} is the total volume of oil-in-place at $t = 0$, which are defined as follows,

$$V_{n_{out}} = L_z \int_0^T \int_0^L \int_0^L -q_n \; dx \; dy \; dt, \tag{8.6}$$

$$V_{OIP} = L_z \int_0^L \int_0^L \phi \cdot (1 - S_w(x, y, 0)) \; dx \; dy = 2.4519 \times 10^6 \text{ ft}^3. \tag{8.7}$$

8.2 Numerical results

The two-phase flow problem described above is solved using the different discretizations listed below, and their output predictions are compared for accuracy and efficiency.

1. Finite volume (FV) method using a two-point flux approximation with upstream mobility weighting, and a backward Euler (BDF1) time-marching scheme. Solved on uniformly spaced quadrilateral meshes.

2. Piecewise linear (P1) DG method using a second order backward differentiation formula (BDF2) time-marching scheme. Solved on uniformly spaced quadrilateral meshes.

3. Piecewise quadratic (P2) DG method using a third order backward differentiation formula (BDF3) time-marching scheme. Solved on uniformly spaced quadrilateral meshes.

4. Piecewise linear (P1) space-time DG method on structured tetrahedral meshes.

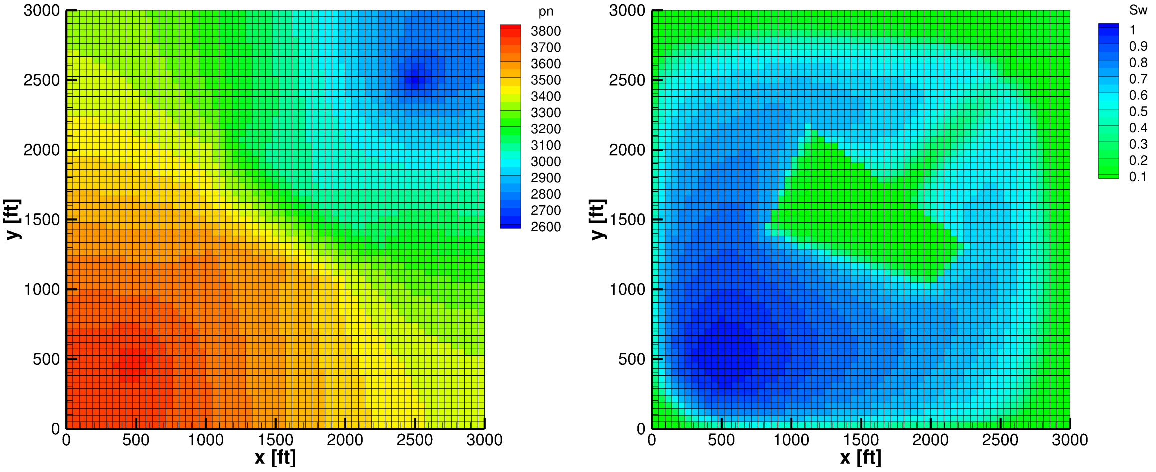
5. Piecewise quadratic (P2) space-time DG method on structured tetrahedral meshes.

6. Piecewise linear (P1) adaptive space-time DG method on unstructured tetrahedral meshes.

7. Piecewise quadratic (P2) adaptive space-time DG method on unstructured tetrahedral meshes.

The first-order finite volume method does not require artificial viscosity since it does not introduce spurious oscillations, and therefore solves the two-phase flow equations directly without any artificial viscosity. In contrast, the DG methods use the space-time artificial viscosity method from Sections 7.2.2 and 7.4, and solve a coupled system of three equations. The DG methods also use the upwinding terms from Chapter 4. Furthermore, all seven methods use the distributed well model introduced in Chapter 5 for specifying the well behavior, with continuity order $m = 6$ and model radius $R = 100$ ft.

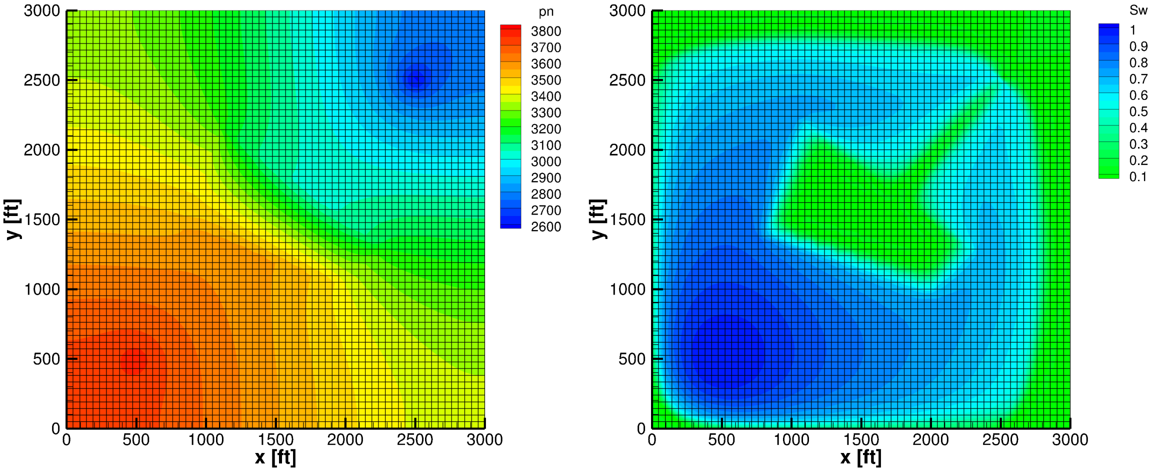
Figures 8-2 shows snapshots of the pressure and saturation solutions at the final time $t = 4000$ days, as obtained from the finite volume method on a 63×63 structured quadrilateral grid with 500 timesteps. The pressure distribution shows significant gradient changes across the boundary of the quadrilateral block where the permeability changes discontinuously, and the saturation front is observed to flow around the low permeability region, with water breakthrough occuring at around



(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 8-2: Primal solutions from the FV method with BDF1 on a 63 × 63 grid with 500 timesteps ($\Delta t = 8$ days), at $t = 4000$ days



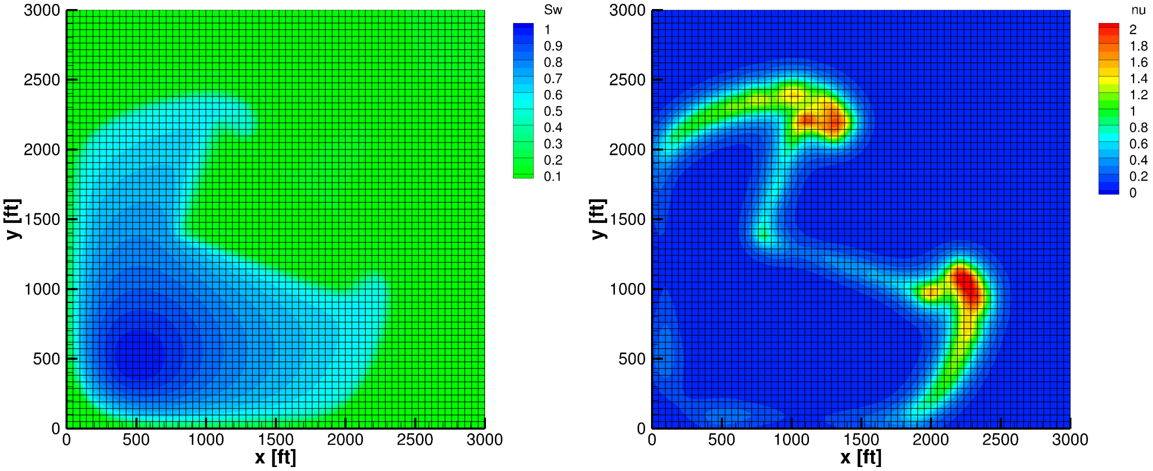
(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 8-3: Primal solutions from the P1 DG method with BDF2 on a 63 × 63 grid with 500 timesteps ($\Delta t = 8$ days), at $t = 4000$ days

$t = 3250$ days. The solutions from the P1 DG method with BDF2 are very similar to the finite volume solutions, as seen in Figure 8-3.

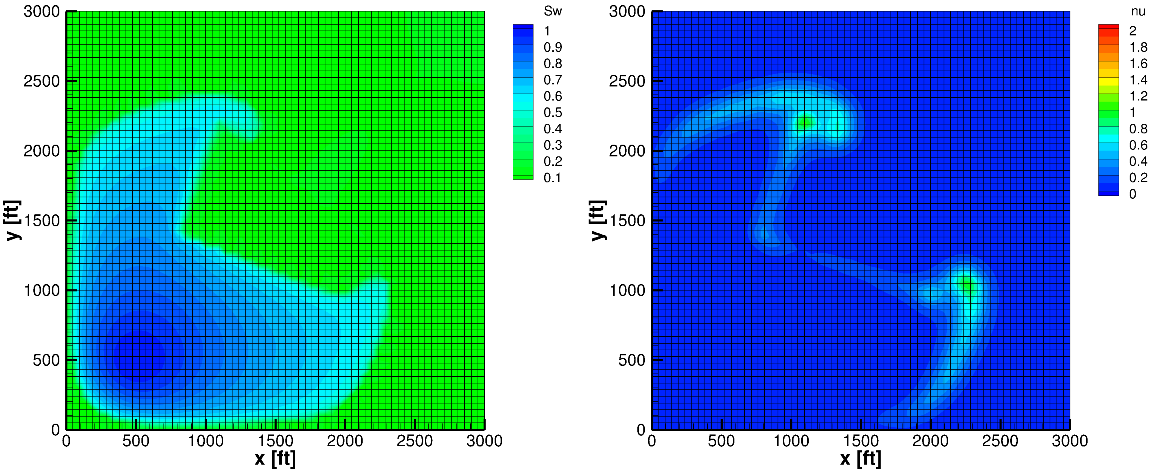
Figures 8-4 and 8-5 show contour plots of the saturation and artificial viscosity at the halfway point (i.e. $t = 2000$ days), as obtained from the P1 and P2 time-marching DG methods respectively. The artificial viscosity is observed to track the saturation front closely in both solutions, but the P2 discretization requires a smaller amount of artificial viscosity on the same mesh due to its greater sub-cell resolution.



(a) Wetting phase saturation S_w

(b) Artificial viscosity ν

Figure 8-4: Primal solutions from the P1 DG method with BDF2 on a 63 × 63 grid with 500 timesteps ($\Delta t = 8$ days), at $t = 2000$ days

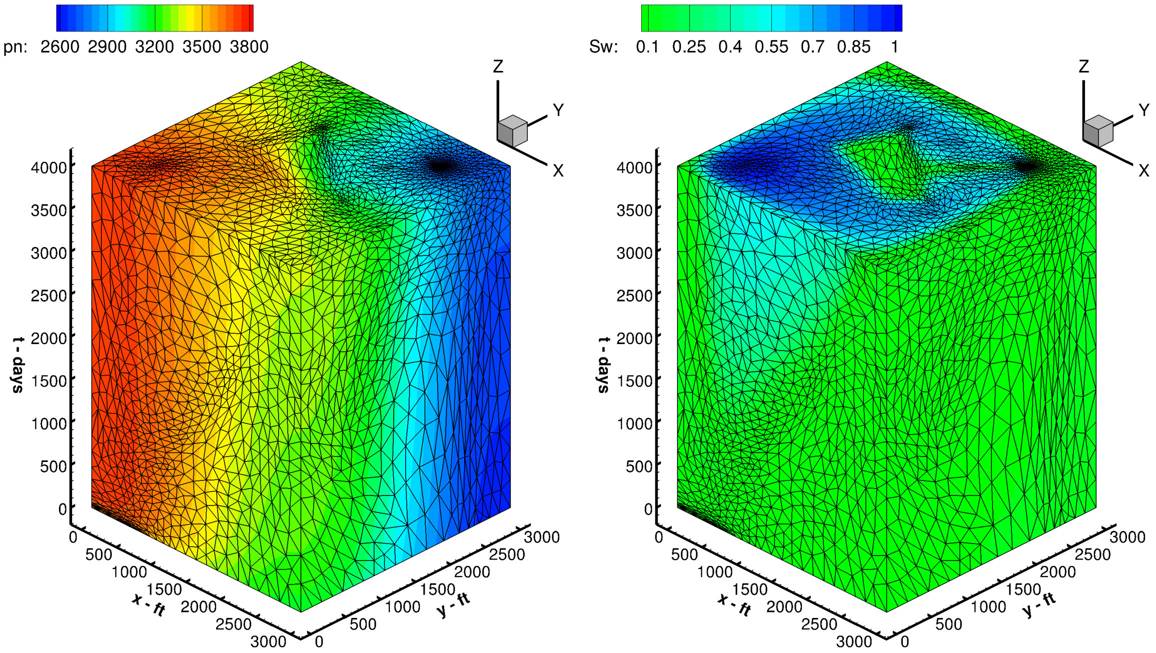


(a) Wetting phase saturation S_w

(b) Artificial viscosity ν

Figure 8-5: Primal solutions from the P2 DG method with BDF3 on a 63 × 63 grid with 500 timesteps ($\Delta t = 8$ days), at $t = 2000$ days

Figure 8-6 shows the final space-time mesh obtained after 20 mesh adaptation iterations, where a P1 DG solution is adapted to a target of 10^6 degrees-of-freedom per primary variable. A crinkled cross-section of this final mesh along the $x = y$ plane is shown in Figures 8-7a and 8-7b, overlaid on the p_n and S_w solutions respectively. The adaptive algorithm focuses on the near-well regions as seen previously in Chapter 6, with more anisotropic resolution around the production well than around the injection well. The algorithm also automatically targets the boundaries of the heterogeneous



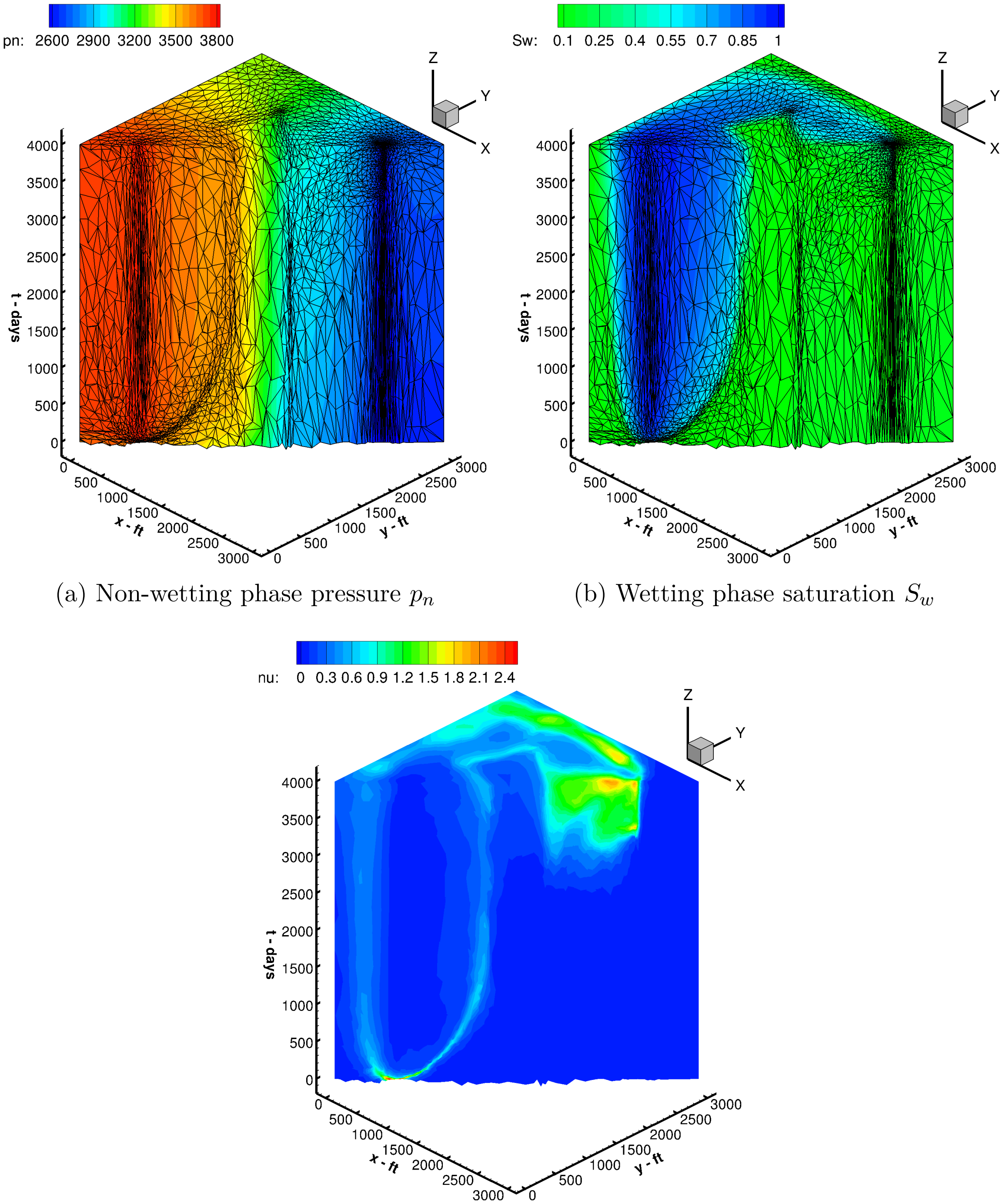
(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

Figure 8-6: Space-time mesh after 20 iterations of the adaptive algorithm, with P1 DG primal solutions adapted to 10^6 DOF

block, without any prior knowledge of its existence, in order to accurate capture the permeability discontinuities. The saturation front is not as sharply resolved as in the homogeneous problem of Chapter 6, but this can be attributed to the fact that the mesh optimization algorithm now has a greater number of important features to resolve using the same DOF budget, and as a result, certain solution features are prioritized over others. Figure 8-7c shows the evolution of artificial viscosity in the space-time domain. Since the artificial viscosity is driven by a sensor that detects inter-element jumps in the saturation variable, artificial viscosity is constantly produced at the moving saturation front, and then spread smoothly in space-time by the diffusion operator in the artificial viscosity PDE. The large values of ν observed at the production well after water breakthrough are due to large pressure gradients near the well, which lead to large flow velocities and thereby large values of $\nu_{\max}$. Although $\nu_{\max}$ is large near the wells at all times, ν itself becomes large only after water breaks through and activates the cell sensor s in that region.

Figures 8-8 to 8-10 show cross-sections of the P1 adapted space-time DG solution



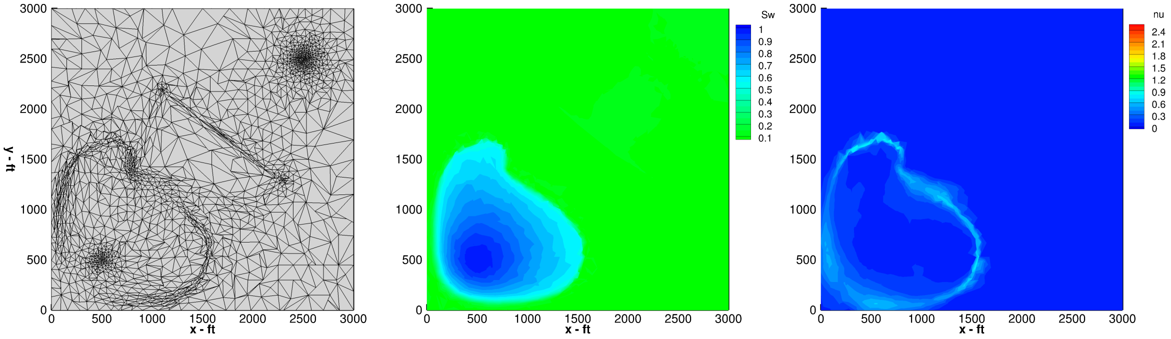
(a) Non-wetting phase pressure p_n

(b) Wetting phase saturation S_w

(c) Artificial viscosity ν

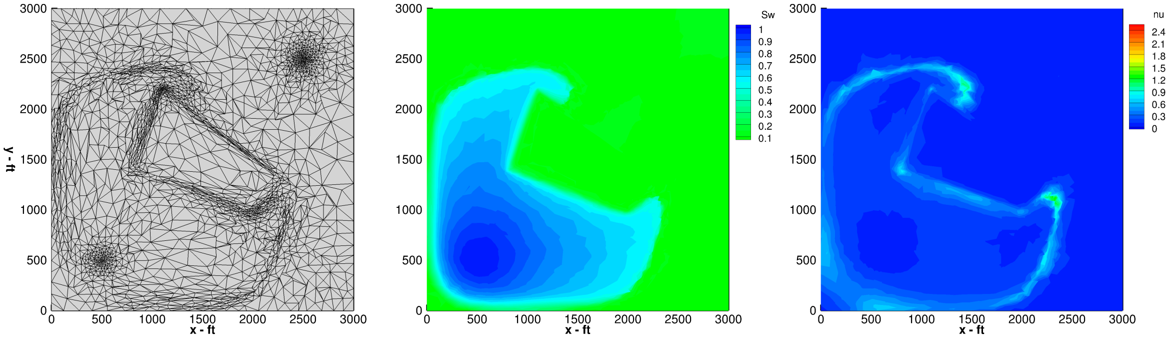
Figure 8-7: Cross-section of final adapted space-time mesh, with P1 DG primal solutions adapted to 10^6 DOF

at $t = 1000, 2000$ and 3000 days. The first column shows the crinkled surface of the space-time mesh sliced at different times, as viewed from directly above the $x{-}y$ plane.



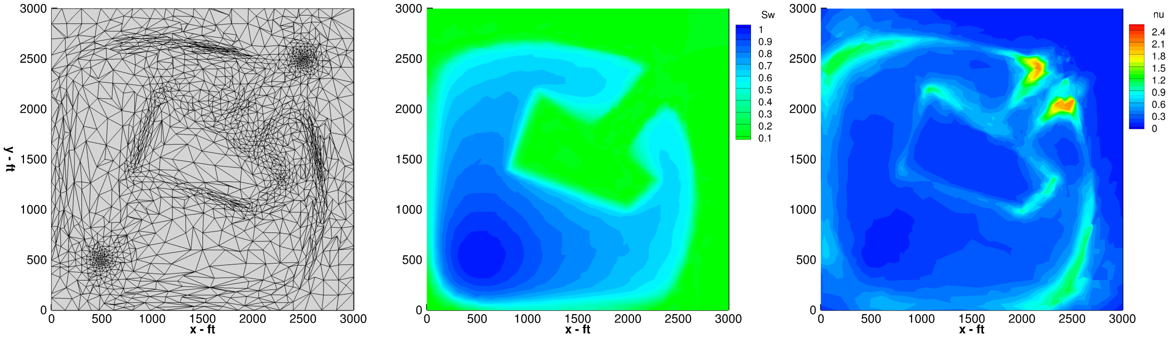
(a) Mesh cross-section (b) Saturation S_w (c) Artificial viscosity ν

Figure 8-8: Slice of P1 space-time DG solution at $t = 1000$ days



(a) Mesh cross-section (b) Saturation S_w (c) Artificial viscosity ν

Figure 8-9: Slice of P1 space-time DG solution at $t = 2000$ days



(a) Mesh cross-section (b) Saturation S_w (c) Artificial viscosity ν

Figure 8-10: Slice of P1 space-time DG solution at $t = 3000$ days

The second and third columns show the distribution of saturation S_w and artificial viscosity ν respectively, at the above times. The figures above clearly show how the spatial resolution of the mesh evolves with time, in order to accurately capture the solution features that are important for the output of interest at a given time.

An accurate representation of geological heterogeneities does not appear to be equally important at all times, as evidenced by the varying levels of resolution allocated to the boundaries of the low permeability region by the adaptive algorithm, at $t = 1000, 2000$ and 3000 days. It is expected that the ability of the adaptive space-time DG method to reallocate degrees-of-freedom between static solution features (e.g. wells, geological heterogeneities) and dynamic solution features (e.g. moving saturation front) will enable it to outperform other numerical methods based on "fixed" meshes.

Figure 8-11 shows the convergence of the oil recovery factor predictions with mesh refinement. In order to make a fair comparison between the FV and DG methods, the abscissa represents the total number of space-time unknowns in each solution, given by $(N_{\text{DOF}} N_{\text{state}})$, where N_{DOF} is the number of space-time degrees-of-freedom per state variable, and N_{state} is the number of state variables used by the discretization. For the finite volume method which only solves for pressure and saturation, $N_{\text{state}} = 2$, whereas for the DG methods $N_{\text{state}} = 3$, since they also solve for artificial viscosity. However, even with the additional degrees-of-freedom associated with artificial viscosity, the P1 and P2 adaptive space-time DG methods (solid red and green lines) only require $\sim 10^6$ unknowns to achieve an error tolerance of $\pm 0.1\%$ (marked by the dotted horizontal lines). In contrast, the time-marching finite volume method requires $\sim 10^9$ unknowns to achieve the same error level. The time-marching DG methods and the space-time DG methods on structured tetrahedral meshes fare inbetween the two extremes discussed above.

The runtime performance of the time-marching and adaptive space-time DG methods is investigated below by comparing the wall-clock times taken to achieve a 0.1% output error level. For each time-marching and adaptive space-time method in Figure 8-11, the case with an error level closest to 0.1% is selected, and executed with 2 to 128 parallel MPI processes. All cases were run on a high performance computing (HPC) cluster where each compute node contained two Intel Xeon E5-2660 v3 (2.6 GHz, 10-core) processors, each with a 25 MB cache. The wall-clock times of each of these simulations are shown in Figures 8-12. The timing data shown for the adaptive space-time DG method represent the total time taken for 20 intermediate mesh adaptation iterations, which include the time taken for the primal solves, adjoint solves, error estimation, and local sampling. In contrast, the data of all the time-marching methods only represent the time taken for a single primal solve.

The time-marching FV method and the P1 adaptive space-time DG method have comparable wall-clock times, especially for larger numbers of MPI processes, and are also the slowest to achieve a 0.1% error level. The P1 time-marching DG method and the P2 adaptive space-time DG method also have comparable wall-clock times, and are roughly 2-3 times faster than the time-marching FV method. The P2 time-marching DG method is about a further 5 times faster. It appears that repeatedly solving the smaller linear systems produced by the time-marching discretizations at each timestep is more runtime efficient compared to solving the larger linear systems produced by the space-time DG discretizations. However, the adaptive space-time DG methods include the time taken for 20 intermediate solves, and also automatically provide additional information such as adjoint solutions and output error estimates, for which the time-marching methods require additional work. The runtime efficiency gains of the high-order DG methods, relative to the FV method, are significantly lower for this heterogeneous problem compared to the results of the homogeneous problem in Chapter 6, primarily due to the costs associated with solving for the additional state variable ν. Figure 8-13 shows the speed-up factor of each method relative to a serial execution for different MPI process counts. As before, the space-time parallelism of the adaptive space-time DG methods enable them to achieve a significantly greater scalability compared to the time-marching methods, which only have spatial parallelism. A more detailed investigation of the wall-clock times of the P1 and P2 adaptive space-time DG methods is presented in Appendix E.

8.3 Summary

In this chapter, the output prediction performance of the adaptive space-time DG method was compared against time-marching and space-time methods using struc-

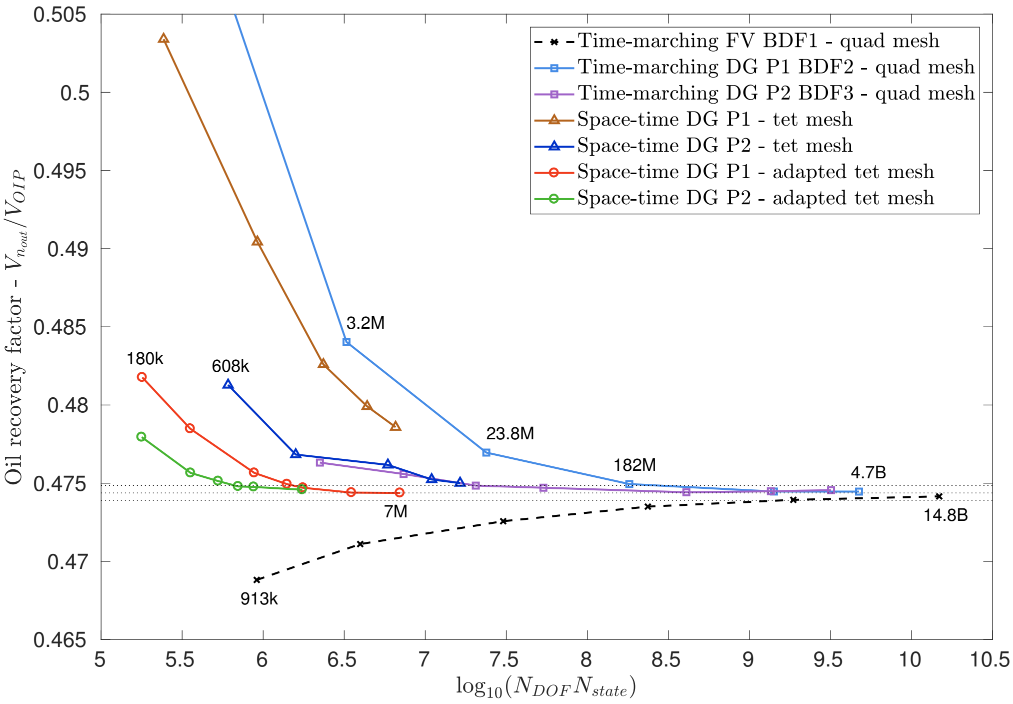


Figure 8-11: Output vs. space-time DOF for different discretizations

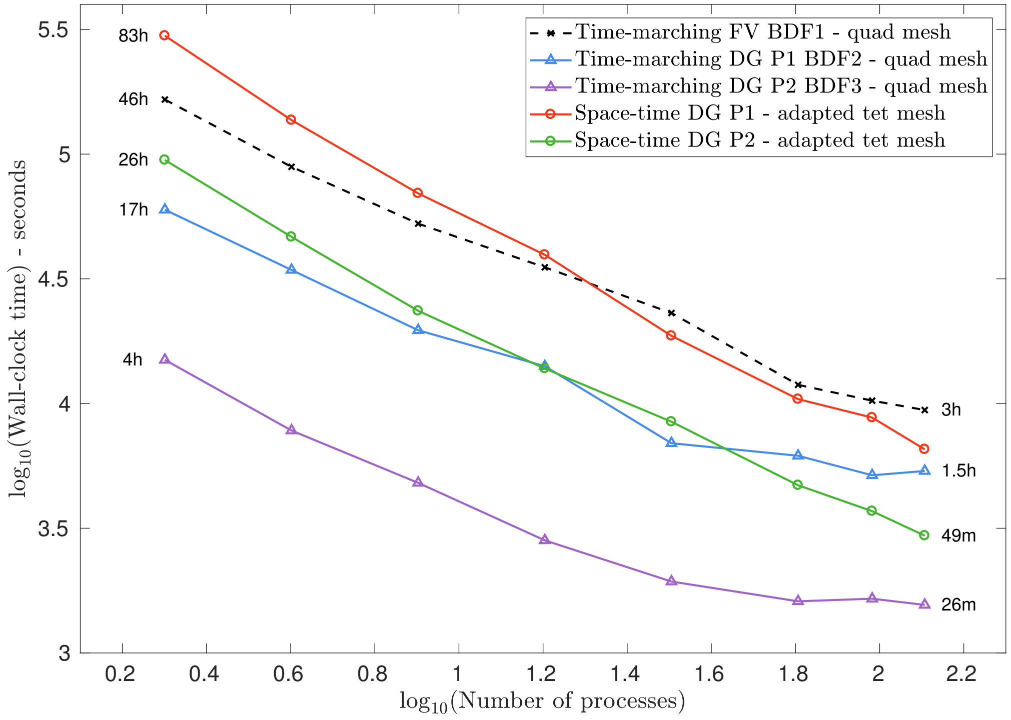


Figure 8-12: Wall time required to achieve 0.1% output error vs. MPI process count

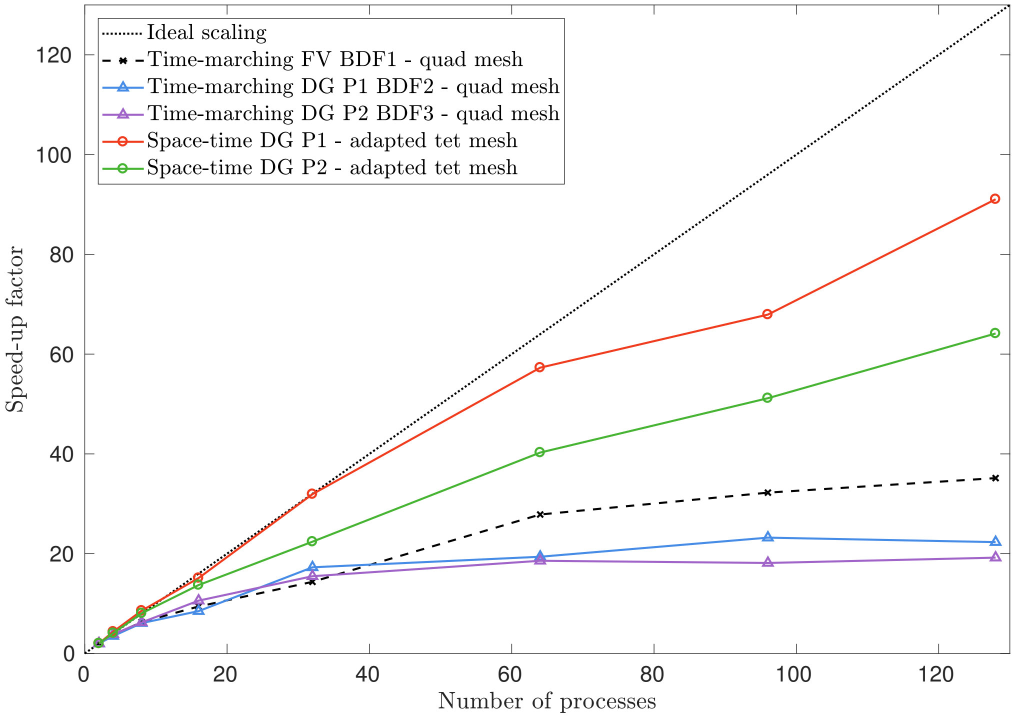


Figure 8-13: Speed-up factor vs. MPI process count

tured meshes, on a slightly compressible two-phase flow problem in a heterogeneous reservoir. Due to the absence of capillary effects, the high-order DG discretizations relied on the space-time artificial viscosity method proposed in Chapter 7 for stability and robustness. Numerical results from the different discretizations were presented, where it was observed that the artificial viscosity in the DG methods closely tracked the saturation front, successfully avoiding the onset of Gibbs phenomenon. Results from the adaptive space-time DG method showed that most of its degrees-of-freedom were spent resolving the near-well regions, the permeability discontinuities, and the saturation front, as expected. A comparison of the oil recovery factor predictions from the different discretizations showed that the P1 and P2 adaptive space-time DG methods were significantly more efficient compared to the other methods considered, in terms of degrees-of-freedom required to achieve a given level of output accuracy. The P2 time-marching DG method required the least amount of wall-clock and CPU-time to achieve a 0.1% error in the output. However, the P1 and P2 adaptive space-time DG methods outperformed the time-marching schemes in terms of parallel scalability.

Chapter 9

Conclusion

9.1 Summary

This thesis presents work towards the development of an efficient and robust numerical framework for solving reservoir flow problems. In particular, a high-order space-time discontinuous Galerkin finite element discretization is coupled with the MOESS mesh adaptation algorithm [137], to reliably predict output quantities of interest in porous media flows. Following the previous work done in [78], where this approach was demonstrated on one-dimensional reservoir flow problems, the focus of this thesis is to extend and apply the solution framework to more realistic problems in higher dimensions.

In order to understand and validate the mesh adaptation behaviors observed on certain reservoir problems, it was necessary to develop a theoretical understanding of the adjoint equations and solution behavior for representative models of porous media flows. This motivated the adjoint analysis work in this thesis, where the adjoint equations and boundary conditions were derived for the Buckley-Leverett and two-phase flow equations. The analytic adjoint solutions derived for the Buckley-Leverett equation also serve as useful reference data for verifying the adjoint consistency of numerical schemes.

One of the major obstacles in extending the adaptive framework to multi-dimensional reservoir flow problems was the lack of a feasible well model for capturing the interaction between the well-bore and the reservoir for finite element methods on unstructured meshes. The discretization-agnostic distributed well model developed in this thesis addressed this issue, and numerical results were presented to show that the flow rate predictions of the proposed well model were not only consistent with the widely used Peaceman well model, but also more robust to changes in the mesh.

A significant portion of this thesis is also aimed at improving the stability and robustness of the DG discretization, particularly for the two-phase flow equations in the advection-dominant limit. Modifications to the BR2 diffusive flux [21, 22] were derived based on a linearized analysis of the analytic and discrete two-phase flow equations, which effectively upwind the underlying saturation equation. The additional interior interface terms produced from this analysis were shown to stabilize the BR2 discretization of a 1D test problem. Later in the thesis, a modified form of Barter's PDE-based artificial viscosity method [17] was proposed for the Buckley-Leverett and two-phase flow equations, as a means of mitigating Gibbs oscillations that usually occur in high-order discretizations, and ensuring convergence to physical solutions.

The adaptive space-time DG method developed in this work was demonstrated on two compressible two-phase flow problems, with homogeneous and heterogenous reservoirs. Comparisons with conventional time-marching methods and space-time DG methods on structured meshes showed that the adaptive space-time DG method is significantly more efficient at predicting output quantities of interest, in terms of degrees-of-freedom required to achieve a given level of accuracy. Studies of wall-clock times also revealed the superior parallel scalability of the adaptive space-time DG method relative to time-marching methods.

With the results of this thesis clearly demonstrating the significant potential of solution adaptive space-time methods, it is hoped that the contributions of this thesis are adopted, improved and extended to more realistic and practical reservoir flow problems.

9.2 Future work

This section contains a few brief discussions on possible areas for further research.

3D spatial domains

The extension of the space-time adaptive framework to 3D spatial problems requires the development of one major tool: a 4D metric-conforming mesh generator. Although there have been a few attempts at 4D unstructured mesh generation [26, 103], none of the methods are known to generate highly anisotropic or metric-conforming meshes. It is speculated that an incremental, local mesh refinement technique based on a modified Delaunay-kernel is a good starting point for developing a metric-conforming 4D simplex mesh generator. Such an approach is presented for 2D meshes in [27] and for 3D meshes in [45]. The dimension-independent, Delaunay-based anisotropic meshing algorithm proposed recently by Caplan et al. [31] also seems to be an attractive option for this work. Extending the local sampling step of the MOESS algorithm to pentatope meshes will also require the solver infrastructure to construct and solve 10 local configurations (10 edge-splits) for each element in the 4D mesh.

Preconditioners

Although the ILU(k) preconditioner with the minimum discarded fill-in (MDF) element ordering [119] is used successfully for the problems in this thesis, its performance on more heterogeneous problems remains to be investigated. Furthermore, it may also be possible to design an efficient, problem-specific preconditioner that exploits the hyperbolicity of the space-time problem in the temporal direction, or the ellipticity of the pressure solution. There exist many preconditioners for the latter case in the context of time-marching problems [132, 29, 2, 94], but their applicability for space-time discretizations requires further study.

Discrete geological models

The geological heterogeneity considered in this work consists of a discontinuous decrease in rock permeability, which was confined to a single analytically-described quadrilateral region of the reservoir. However, in practical reservoir simulation problems, the geological data is usually provided as node-averaged or cell-averaged values on a discrete grid. For large scale reservoirs, the geological models are often defined on finer meshes compared to the meshes used by the flow solver, thereby requiring the geological data to be upscaled before it can be used in the simulation. This is especially true for the adaptive method, since the mesh adaptation process usually starts with significantly coarse meshes. In order to be able to work with such discrete geological models, the adaptive method requires the implementation of an appropriate "sampling" process, where the reservoir properties at any given point in the unstructured space-time mesh can be evaluated from the data in the fine-scale geological model. A simple approach would be to perform an L^2 projection of the geological data on to each adapted space-time mesh, thus reducing the data to a piecewise polynomial representation that can be evaluated at each quadrature point by the finite element method. However, this approach unnecessarily replicates the spatial data in the temporal direction, and therefore requires further investigation for improving data storage efficiency. The waterflooding problem of the 10th SPE Comparative Solution Project (SPE10) [37] is proposed as a suitable test case for studying the performance of the adaptive space-time DG method on problems with discrete geological models in the future.

Appendix A

Cost analysis of the space-time DG mesh adaptation framework

In the interest of fairly comparing computational costs between different spatial dimensions d, the mesh resolution is assumed to be fixed and characterized by an average mesh size $h < 1$ in each dimension, with the volume of the space-time domain being equal to 1. The number of $(d+1)$-simplex elements in the space-time mesh is given by,

$$N_{\text{elem}} \sim \frac{1}{V_{\text{simplex}}(d+1) \cdot h^{d+1}}, \tag{A.1}$$

where $V_{\text{simplex}}(d)$ represents the volume of a unit d-dimensional regular simplex,

$$V_{\text{simplex}}(d) = \frac{1}{d!}\sqrt{\frac{d+1}{2^d}}. \tag{A.2}$$

For a space-time DG discretization, the number of degrees of freedom per element, M, is given by:

$$M(d,p) = \frac{(p+d+1)!}{p!\ (d+1)!}\ N_{\text{state}}, \tag{A.3}$$

where p is the polynomial order of the DG solution and N_{state} is the number of state variables. Therefore, the total number of degrees of freedom on a space-time mesh with N_{elem} elements is given by:

$$N(d,p) = M(d,p) \cdot N_{\text{elem}}, \tag{A.4}$$

For simplicity, it is assumed that the complexity of the linear solver can be modeled as $\mathcal{O}(kM^3N_{\text{elem}}^r)$, where k is a constant that captures the nonlinearity and the conditioning of the physical problem. Highly nonlinear, poorly conditioned problems result in larger k values. The cubic power on M is a result of the dense coupling of DOFs within each element, which produces a dense matrix block of size $M \times M$ that needs to be solved for each element. However, since the DG discretization has sparse interactions between elements, the Jacobian matrix has a block sparse structure that can be exploited by sparse matrix solvers to produce a more efficient scaling on N_{elem}. Thus, the exponent r typically takes values in the range $1 \leq r \leq 2$ for well preconditioned iterative solvers of sparse systems. Therefore, the cost of solving the primal problem is given by:

$$C_{\text{primal}} = \mathcal{O}\left(k_{\text{primal}} \cdot (M(d,p))^3 \cdot (N_{\text{elem}})^r\right). \tag{A.5}$$

Similarly, the cost of solving the adjoint problem in the richer $(p+1)$ space is:

$$\begin{aligned} C_{\text{adjoint}} &= \mathcal{O}\left(k_{\text{adjoint}} \cdot (M(d,p+1))^3 \cdot (N_{\text{elem}})^r\right) && \text{(A.6)} \\ &= \mathcal{O}\left(k_{\text{adjoint}} \cdot \left(M(d,p) \cdot \frac{p+d+2}{p+1}\right)^3 \cdot (N_{\text{elem}})^r\right) && \text{(A.7)} \\ \frac{C_{\text{adjoint}}}{C_{\text{primal}}} &\sim \frac{k_{\text{adjoint}}}{k_{\text{primal}}} \cdot \left(\frac{p+d+2}{p+1}\right)^3. && \text{(A.8)} \end{aligned}$$

Although the cubic term in Eq. (A.8) suggests that the adjoint solve may be more expensive compared to the primal solve, the linearity of the adjoint problem compensates for this via the $k_{\text{adjoint}}/k_{\text{primal}}$ ratio, often making the adjoint solve cheaper than solving the nonlinear primal problem. The relative cost of the adjoint solve further diminishes with increasing p.

In the local sampling step of the MOESS algorithm, each local configuration is obtained by splitting an edge of an element to produce two sub-elements. The cost of performing a single local solve to compute the solution on these two sub-elements is given by:

$$C_{\text{config}} = \mathcal{O}\left(k_{\text{local}} \cdot (M(d,p))^3\right). \tag{A.9}$$

Therefore, the cost of all local solves is given by:

$$C_{\text{local}} = C_{\text{config}} \cdot N_{\text{config}} \cdot N_{\text{elem}}, \tag{A.10}$$

where N_{config} is the number of split configurations per element, which is equal to the number of edges in the $(d+1)$-simplex element:

$$N_{\text{config}} = \frac{1}{2}(d+1)(d+2). \tag{A.11}$$

Hence, the cost of local solves simplifies to:

$$\begin{aligned} C_{\text{local}} &= \mathcal{O}\left(k_{\text{local}} \cdot (M(d,p))^3\right) \cdot \frac{1}{2}(d+1)(d+2) \cdot N_{\text{elem}} && \text{(A.12)} \\ &= \mathcal{O}\left(k_{\text{local}} \cdot (M(d,p))^3 \cdot (N_{\text{elem}})^r\right) \cdot (d+1)(d+2) \cdot (N_{\text{elem}})^{1-r} && \text{(A.13)} \\ \frac{C_{\text{local}}}{C_{\text{primal}}} &\sim \frac{k_{\text{local}}}{k_{\text{primal}}} \cdot (d+1)(d+2) \cdot \left(V_{\text{simplex}}(d+1) \cdot h^{d+1}\right)^{r-1} && \text{(A.14)} \end{aligned}$$

If $r > 1$, the exponential decrease of $V_{\text{simplex}}(d+1) \cdot h^{d+1}$ with d ensures that the cost of all the local solves is cheaper compared to the primal solve at higher dimensions. Furthermore, each of the local problems are generally less nonlinear and better conditioned than the global problem. This is also observed in practice, where the primal solve typically takes $\mathcal{O}(10)$ nonlinear iterations whereas each local solve takes only $\mathcal{O}(1)$ nonlinear iterations. Therefore, even for the worst-case of $r = 1$ and $d = 3$, the $(d+1)(d+2)$ factor in Eq. (A.14) can be compensated by the $k_{\text{local}}/k_{\text{primal}}$ ratio, making the local sampling procedure cheaper than the primal solve.

The cost of generating a space-time mesh may be approximated by the complexity of computing a Delaunay triangulation of n points in $d+1$ dimensions, which is known to be $\mathcal{O}(n)$ in the expected case and $\mathcal{O}(n^{\lceil (d+1)/2 \rceil})$ in the worst-case [5, 49]. The relationship between n and N_{elem} for an isotropic space-time mesh can be approximated by the following relation:

$$n \approx \xi(d+1) \cdot N_{\text{elem}}, \tag{A.15}$$

where $\xi(d)$ is ratio between the number of vertices per d-simplex and the average number of regular d-simplices around a vertex, $S(d)$,

$$\xi(d) = \frac{d+1}{S(d)}. \tag{A.16}$$

Under the assumption of an isotropic mesh, a good approximation for $S(d)$ is the ratio:

$$S(d) \approx \frac{\Theta_{\text{sphere}}(d)}{\Theta_{\text{vertex}}(d)}, \tag{A.17}$$

where $\Theta_{\text{sphere}}(d)$ is the solid angle subtended by the surface of the d-dimensional unit ball at the origin, and $\Theta_{\text{vertex}}(d)$ is the solid angle subtended by a face of the regular d-simplex at its opposite vertex. Using the formula for the solid angular content at each vertex of a regular simplex given in [90], the above ratio can be written as:

$$\frac{\Theta_{\text{sphere}}(d)}{\Theta_{\text{vertex}}(d)} = \frac{2^d}{d! \; F_d\left(\frac{1}{2}\sec^{-1}(d)\right)}, \tag{A.18}$$

where $F_d(\alpha)$ is the recursive Schläfli function defined in Section 7.2 of [142]. Table A.1 contains evaluations of the above ratios up to $d = 4$.

By assuming the expected linear complexity of the Delaunay triangulation, the cost

d	$S(d)$	$\xi(d)$
1	2	1
2	6	0.5
3	22.795	0.175
4	102.200	0.049

Table A.1: Numerical values of $S(d)$ and $\xi(d)$

of mesh generation relative to the primal solve is given by:

$$\begin{aligned} C_{\text{mesh}} &= \mathcal{O}\left(\xi(d+1) \cdot N_{\text{elem}}\right) && \text{(A.19)} \\ \frac{C_{\text{mesh}}}{C_{\text{primal}}} &\sim \frac{\xi(d+1)}{k_{\text{primal}} \cdot (M(d,p))^3 \cdot (N_{\text{elem}})^{r-1}} && \text{(A.20)} \end{aligned}$$

$$\frac{C_{\text{mesh}}}{C_{\text{primal}}} \sim \frac{\xi(d+1) \cdot \left(V_{\text{simplex}}(d+1) \cdot h^{d+1}\right)^{r-1}}{k_{\text{primal}} \cdot (M(d,p))^3} \qquad \text{(A.21)}$$

If $r > 1$, the decrease of the h^{d+1} term in Eq. (A.21) dominates (since $h < 1$), and causes C_{mesh} to be smaller relative to C_{primal} as d increases. The ratio $\xi(d{+}1)/M(d,p)^3$ also decreases with increasing d. Thus, even for an optimally scaling primal solver (i.e., $r = 1$), the mesh generation cost is a decreasing fraction of the primal solve cost as d increases.

Appendix B

Linearization of the weighted residual

In order to simplify Eq. (3.16) further, consider the expression inside the brackets in the last integral of Eq. (3.16). Expanding out all components of space-time fluxes and normal vectors, and rewriting the expression in terms of 1D jump operators, yields,

$$\xi = \left[\!\left[\frac{\partial \vec{F}}{\partial x} \delta x_s \right]\!\right] + \left(\vec{F}^{+} - \vec{F}^{-} \right) \cdot \delta \vec{n}^{+} \tag{B.1}$$

$$= \left[\!\left[\left(\frac{\partial F_x}{\partial x} n_x + \frac{\partial F_t}{\partial x} n_t \right) \delta x_s \right]\!\right] + [\![F_x]\!]\, \delta n_x^{+} + [\![F_t]\!]\, \delta n_t^{+}, \tag{B.2}$$

where δn_x^{+} and δn_t^{+} are the spatial and temporal components of the perturbed unit normal vector of Γ_s. Since δx_s is unique for both sides of the shock, it is moved out of the jump operator, and the $\frac{\partial F_x}{\partial x}$ term is replaced with $-\frac{\partial F_t}{\partial t}$ using the space-time primal equation in Eq. (3.5), giving,

$$\xi = \left(\left[\!\left[\frac{\partial F_x}{\partial x} \right]\!\right] + \left[\!\left[\frac{\partial F_t}{\partial x} \right]\!\right] \frac{n_t^{+}}{n_x^{+}} \right) n_x^{+} \delta x_s + [\![F_x]\!]\, \delta n_x^{+} + [\![F_t]\!]\, \delta n_t^{+} \tag{B.3}$$

$$= \left(\left[\!\left[-\frac{\partial F_t}{\partial t} - \dot{x}_s \frac{\partial F_t}{\partial x} \right]\!\right] \right) n_x^{+} \delta x_s + [\![F_x]\!]\, \delta n_x^{+} + [\![F_t]\!]\, \delta n_t^{+}. \tag{B.4}$$

Following the approach used in [67] and using the definition $\frac{d(\cdot)}{dt} = \frac{\partial(\cdot)}{\partial t} + \dot{x}_s \frac{\partial(\cdot)}{\partial x}$, yields,

$$\frac{d\,[\![F_t]\!]}{dt} = \left[\!\!\left[\frac{\partial F_t}{\partial t} + \dot{x}_s \frac{\partial F_t}{\partial x}\right]\!\!\right], \tag{B.5}$$

and therefore,

$$\xi = -\frac{d\,[\![F_t]\!]}{dt} n_x^+ \delta x_s + [\![F_x]\!]\,\delta n_x^+ + [\![F_t]\!]\,\delta n_t^+. \tag{B.6}$$

Expanding the space-time jump condition in Eq. (3.7) gives,

$$[\![F_x]\!] = -\,[\![F_t]\!]\,\frac{n_t^+}{n_x^+}, \tag{B.7}$$

which is substituted in Eq. (B.6) to obtain:

$$\xi = -\frac{d\,[\![F_t]\!]}{dt} n_x^+ \delta x_s - [\![F_t]\!]\left(\frac{n_t^+ \delta n_x^+}{n_x^+} - \delta n_t^+\right). \tag{B.8}$$

The final step requires a relationship between δx_s and the components of the perturbed unit normal vector $\vec{n}^+$. This is derived by linearizing the ratio of n_t^+/n_x^+ as defined in Eq. (3.10),

$$\dot{x}_s = -\frac{n_t^+}{n_x^+} \tag{B.9}$$

$$\dot{x}_s + \delta\dot{x}_s = -\frac{n_t^+ + \delta n_t^+}{n_x^+ + \delta n_x^+}. \tag{B.10}$$

Eq. (B.10) is simplified further using a Taylor series expansion of the right-hand side and retaining the linear terms as follows,

$$\dot{x}_s + \delta\dot{x}_s = -\left(n_t^+ + \delta n_t^+\right)\left(\frac{1}{n_x^+} - \frac{\delta n_x^+}{n_x^{2+}} + \mathcal{O}(\delta n_x^{2+})\right) \tag{B.11}$$

$$\delta\dot{x}_s = \frac{n_t^+ \delta n_x^+}{n_x^{2+}} - \frac{\delta n_t^+}{n_x^+}. \tag{B.12}$$

Noting that the right-hand side of Eq. (B.12) appears inside the brackets of the last term in Eq. (B.8), the expression for ξ is finally given by,

$$\begin{aligned}\xi &= -\frac{d\,[\![F_t]\!]}{dt} n_x^+ \delta x_s - [\![F_t]\!]\, \delta \dot{x}_s n_x^+ && \text{(B.13)}\\ &= -\frac{d}{dt}\left([\![F_t]\!]\, \delta x_s\right) n_x^+ && \text{(B.14)}\\ &= -\frac{d}{dt}\Big((F_t^+ - F_t^-)\delta x_s\Big)\, n_x^+ . && \text{(B.15)}\end{aligned}$$

Appendix C

Output sensitivities via adjoint solutions

This section outlines how solutions to the continuous adjoint problem can be used to compute the sensitivities of an output functional to parameters in the model. Assume that the weak form residual equation which needs to be satisfied by the primal solution $u(x,t,\alpha)$ is given by,

$$R(u,w,\alpha) = 0, \tag{C.1}$$

where $w(x,t)$ is an admissible test function and α is a model parameter. The weak form residual R may be nonlinear in u and α, but it is linear in w. The linearized form of the above equation is obtained by considering infinitesimal perturbations of the model parameter, $\delta\alpha$, which also results in perturbations of the primal solution, δu. Linearizing the weak form residual equation with the above perturbations about a primal solution $u(x,t,\alpha)$ yields,

$$\begin{aligned} R(u+\delta u, w, \alpha+\delta\alpha) &= 0, \\ R(u,w,\alpha) + \delta R(\delta u, w, \delta\alpha) &= 0, \\ \delta R(\delta u, w, \delta\alpha) &= 0. \end{aligned} \tag{C.2}$$

Furthermore, the δR constraint in Eq. (C.2) can be expanded as,

$$\delta R(\delta u, w, 0) + \delta R(0, w, \delta\alpha) = 0. \tag{C.3}$$

For a given generic output function $J(u, \alpha)$, the adjoint solution $\psi(x, t)$ satisfies the following equation for all δu,

$$\delta R(\delta u, \psi, 0) = \delta J(\delta u, 0). \tag{C.4}$$

The total perturbation in the output is given by $\delta J(\delta u, \delta\alpha)$, which can be decomposed and re-written using Eq. (C.4) and (C.3) as,

$$\begin{aligned} \delta J(\delta u, \delta\alpha) &= \delta J(\delta u, 0) + \delta J(0, \delta\alpha) \\ &= -\delta R(0, \psi, \delta\alpha) + \delta J(0, \delta\alpha). \end{aligned} \tag{C.5}$$

Note that the absence of δu in the right-hand side of Eq. (C.5) allows the output perturbation to be evaluated directly from $\delta\alpha$ without first calculating δu. Therefore, this adjoint-based sensitivity method is more efficient than the direct method when multiple sensitivity evaluations are required.

Appendix D

Adjoint relationships between equivalent sets of PDEs

Consider the following linearized primal problem,

$$Lu = f, \qquad \text{in } \Omega, \tag{D.1}$$

$$Bu = e, \qquad \text{on } \Gamma, \tag{D.2}$$

where $u \in \mathbb{R}^n$ is the primal solution vector, $L : \mathbb{R}^n \to \mathbb{R}^n$ is a linear differential operator in the domain $\Omega \in \mathbb{R}^d$, and $B : \mathbb{R}^n \to \mathbb{R}^n$ represents the primal boundary condition operator on $\Gamma \in \mathbb{R}^{d-1}$.

The following notation for volume and boundary inner products,

$$(u, v) = \int_\Omega u^T v \; d\Omega, \tag{D.3}$$

$$\langle u, v \rangle = \int_\Gamma u^T v \; d\Gamma, \tag{D.4}$$

allows the output of interest to be written as

$$J = (g, u) + \langle g_B, u \rangle. \tag{D.5}$$

The duality condition produces the following relationship,

$$(g, u) + \langle g_B, u \rangle = (\psi, f) + \langle C^* \psi, e \rangle, \tag{D.6}$$

which is used to derive the corresponding dual problem,

$$L^* \psi = g, \qquad \text{in } \Omega, \tag{D.7}$$

$$B^* C^* \psi = g_B, \qquad \text{on } \Gamma, \tag{D.8}$$

where the adjoint operators L^*, B^* and C^* are derived using integration by parts, as described for the porous media model equations in Chapter 3.

Next, consider an equivalent set of primal equations defined by the transformation matrices $M \in \mathbb{R}^{n \times n}$ and $H \in \mathbb{R}^{n \times n}$:

$$\hat{u} = Mu, \tag{D.9}$$

$$\hat{f} = Hf. \tag{D.10}$$

Rewriting the primal problem in terms of these new quantities yields,

$$\hat{L}\hat{u} = \hat{f}, \qquad \text{in } \Omega, \tag{D.11}$$

$$\hat{B}\hat{u} = e, \qquad \text{on } \Gamma, \tag{D.12}$$

where $\hat{L} = HLM^{-1}$ and $\hat{B} = BM^{-1}$. Similarly, rewriting the output functional gives $J = (\hat{g}, \hat{u}) + \langle \hat{g}_B, \hat{u} \rangle$, where $\hat{g} = M^{-T} g$ and $\hat{g}_B = M^{-T} g_B$.

The duality condition for the transformed problems is manipulated as follows,

$$(\hat{g}, \hat{u}) + \langle \hat{g}_B, \hat{u} \rangle = (\hat{\psi}, \hat{f}) + \langle \hat{C}^* \hat{\psi}, e \rangle \tag{D.13}$$

$$(M^{-T} g, Mu) + \langle M^{-T} g_B, Mu \rangle = (\hat{\psi}, Hf) + \langle \hat{C}^* \hat{\psi}, e \rangle \tag{D.14}$$

$$(g, u) + \langle g_B, u \rangle = (H^T \hat{\psi}, f) + \langle \hat{C}^* \hat{\psi}, e \rangle \tag{D.15}$$

Comparing the volume integrals of Eq. (D.6) and Eq. (D.15) gives the relationship between the adjoint variables of the original primal problem and those of the transformed primal problem:

$$\hat{\psi} = H^{-T}\psi. \tag{D.16}$$

Further, comparing the boundary integrals yields $\hat{C}^* = C^* H^T$. Note that the adjoint variable transformation given in Eq. (D.16) is independent of the solution variable transformation matrix M.

Appendix E

Timing data for adapted cases

This appendix presents a detailed breakdown of the wall-clock times of the adaptive space-time DG method for the heterogeneous two-phase flow problem in Chapter 8.

Figures E-1 and E-2 show the wall-clock times taken by the P1 and P2 adaptive space-time DG methods respectively, to complete 20 mesh adaptation iterations using 16 parallel MPI processes. The different bars on each figure represent results for different target costs, ranging from 50,000 to 400,000 DOFs per state variable. Each bar is subdivided vertically into blue, green and yellow regions to show the proportion of the total time required for the primal solves, adjoint solves, and the local sampling procedures respectively. The time taken for error estimation, mesh generation and solving the mesh optimization problem are ignored since they only constitute a minor fraction of the total time. The percentage labels in each bar show the proportion of time taken by each of the three main steps, as a fraction of the height of the bar. As the size of the problem increases, it is observed that the time-fraction of the primal and adjoint solves increases, whereas the time-fraction of the local sampling procedure decreases.

Similarly, Figures E-3 and E-4 show the breakdown of wall-clock times for a P1 space-time DG method adapted to 400,000 DOF, and a P2 space-time DG method adapted to 150,000 DOF respectively. The different bars on these figures represent results from parallel runs using 2 to 128 MPI processes. The local error sampling times are observed to scale almost perfectly with increasing process count, since the sampling process is embarassingly parallel. In contrast, the time-fraction of the primal solves is observed to increase with the number of processes.

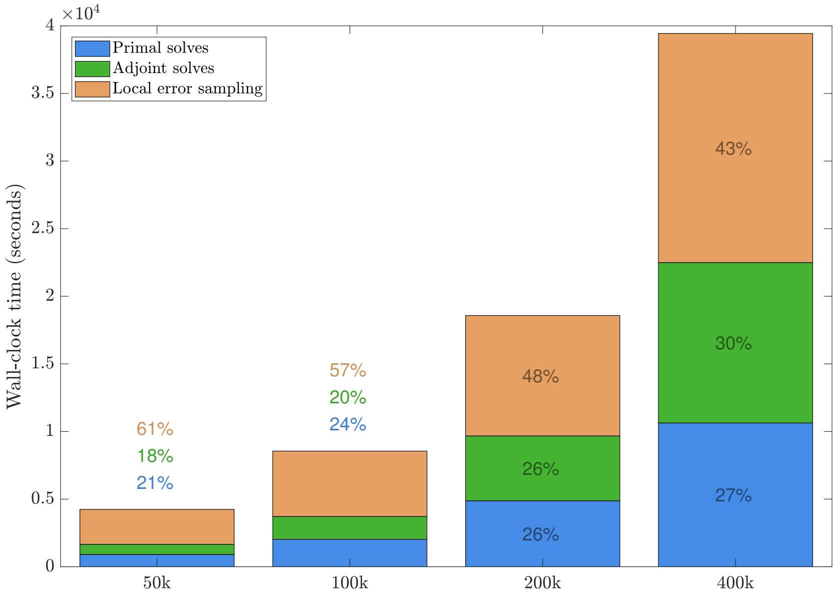


Figure E-1: Breakdown of wall-clock times of the P1 adaptive space-time DG method using 16 MPI processes, for different target DOFs

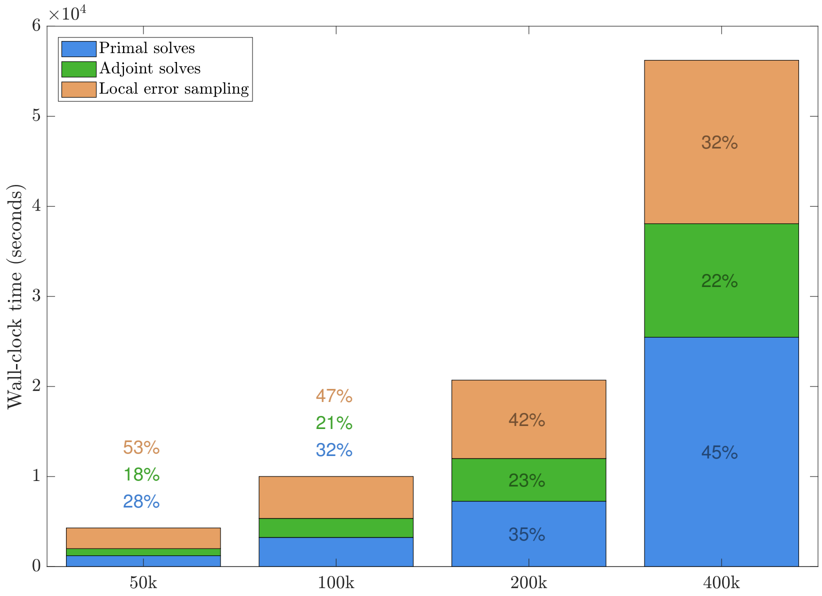


Figure E-2: Breakdown of wall-clock times of the P2 adaptive space-time DG method using 16 MPI processes, for different target DOFs

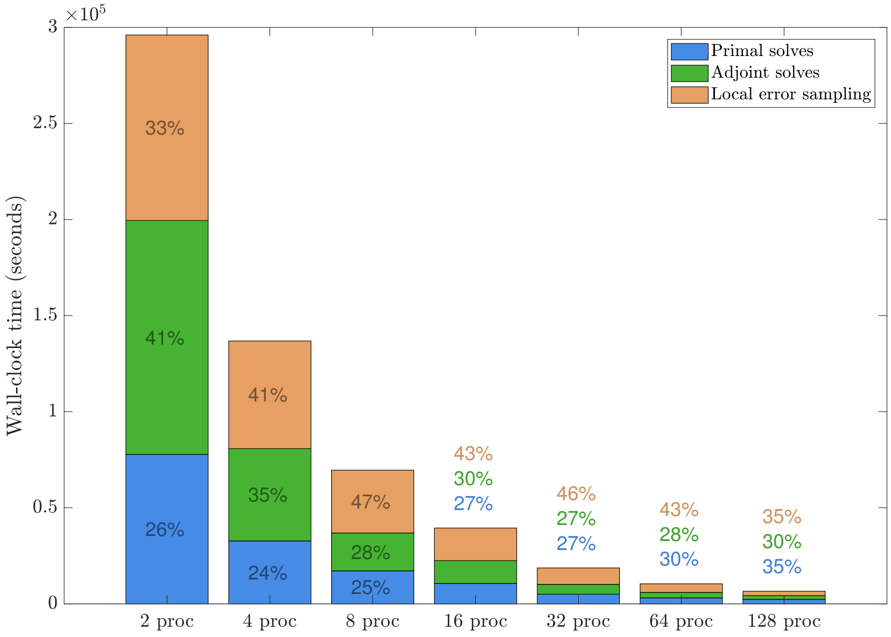


Figure E-3: Breakdown of wall-clock times of the P1 adaptive space-time DG method with 400k DOF, for different MPI process counts

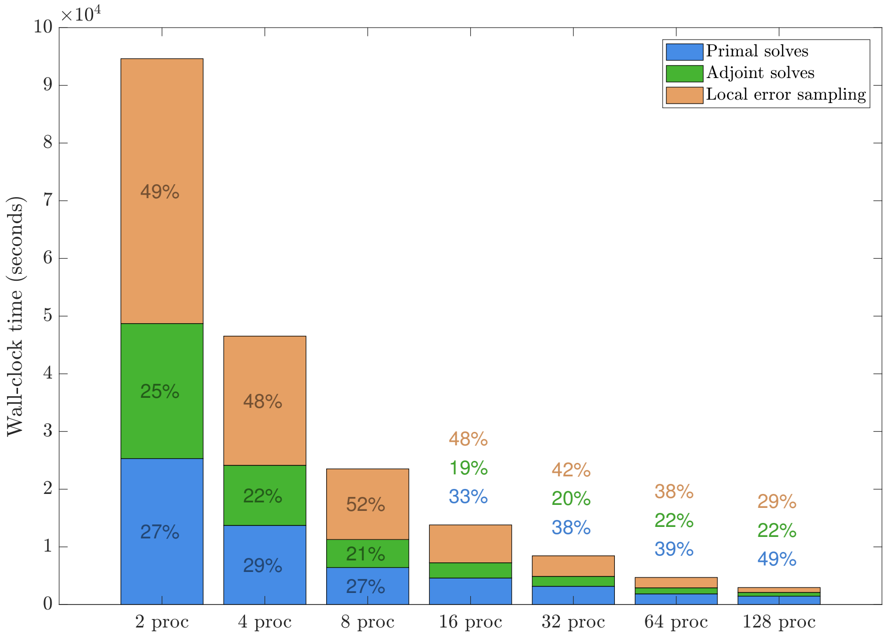


Figure E-4: Breakdown of wall-clock times for the P2 adaptive space-time DG method with 150k DOF, for different MPI process counts

Bibliography

[1] Jamal H. Abou-Kassem and Khalid Aziz. Analytical Well Models for Reservoir Simulation. *Society of Petroleum Engineers*, 35(4), 1985.

[2] Tareq M. Al-Shaalan, Hector Manuel Klie, Ali H. Dogru, and Mary Fanett Wheeler. Studies of Robust Two Stage Preconditioners for the Solution of Fully Implicit Multiphase Flow Problems. *Society of Petroleum Engineers*, 2009.

[3] F. Alauzet and O. Pironneau. Continuous and discrete adjoints to the Euler equations for fluids. *International Journal for Numerical Methods in Fluids*, 70(2):135–157, 2012.

[4] Amaziane, Brahim, Bourgeois, Marc, and El Fatini, Mohamed. Adaptive Mesh Refinement for a Finite Volume Method for Flow and Transport of Radionuclides in Heterogeneous Porous Media. *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles*, 69(4):687–699, 2014.

[5] Nina Amenta and Dominique Attali. Abstract complexity of Delaunay triangulation for points on lower-dimensional polyhedra, 2007.

[6] Todd Arbogast, Mika Juntunen, Jamie Pool, and Mary F. Wheeler. A discontinuous Galerkin method for two-phase flow in a porous medium enforcing H(div) velocityand continuous capillary pressure. *Computational Geosciences*, 17(6):1055–1078, Dec 2013.

[7] J.H. Argyris and D.W. Scharpf. Finite elements in time and space. *Nuclear Engineering and Design*, 10(4):456 – 464, 1969.

[8] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19:742–760, 1982.

[9] Abeeb A. Awotunde and Roland Horne. An Improved Adjoint-Sensitivity Computation for Multiphase Flow Using Wavelets. *Society of Petroleum Engineers*, 2012.

[10] K. Aziz and A. Settari. *Petroleum Reservoir Simulation.* Elsevier, 1979.

[11] D.K. Babu and Aziz S. Odeh. Productivity of a Horizontal Well. *Society of Petroleum Engineers*, 4(4), 1989.

[12] Ivo Babuska, Barna A Szabo, and I Norman Katz. The p-version of the finite element method. *SIAM Journal on Numerical Analysis*, 18(3):515–545, 1981.

[13] T. J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements Anal. Design*, 25:243–273, 1997.

[14] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.

[15] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2017.

[16] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[17] G.E. Barter and D.L. Darmofal. Shock capturing with higher-order, PDE-based artificial viscosity. AIAA 2007-3823, 2007.

[18] Timothy J. Barth and Mats G. Larson. A posteriori error estimates for higher order Godunov finite volume methods on unstructured meshes. In R. Herban and D. Kröner, editors, *Finite Volumes for Complex Applications III*, London, 2002. Hermes Penton.

[19] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous Galerkin solution of the Reynolds averaged Navier-Stokes and k-ω turbulence model equations. *Computers & Fluids*, 34:507–540, May-June 2005.

[20] F. Bassi and S. Rebay. *Accurate 2D Euler computations by means of a high order discontinuous finite element method*, pages 234–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[21] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, *Discontinuous Galerkin Methods: Theory, Computation and Applications*, pages 197–208. Springer, Berlin, 2000.

[22] F. Bassi and S. Rebay. Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 40:197–207, 2002.

[23] Peter Bastian. A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure. *Computational Geosciences*, 18(5):779–796, Oct 2014.

[24] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West Journal of Numerical Mathematics*, 4:237–264, 1996.

[25] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In A. Iserles, editor, *Acta Numerica*. Cambridge University Press, 2001.

[26] Marek Behr. Simplex space-time meshes in finite element simulations. *International Journal for Numerical Methods in Fluids*, 57(9):1421–1434, 7 2008.

[27] Houman Borouchaki, Paul Louis George, Frédéric Hecht, Patrick Laug, and Eric Saltel. Delaunay mesh generation governed by metric specifications. Part I algorithms. *Finite Elements in Analysis and Design*, 25(1-2):61–83, 1997.

[28] Xiao-Chuan Cai and Marcus Sarkis. Restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[29] H. Cao, H.A. Tchelepi, J.R. Wallis, and H.E. Yardumian. Parallel Scalable Unstructured CPR-Type Linear Solver for Reservoir Simulation. *Society of Petroleum Engineers*, 2005.

[30] Weiming Cao, Weizhang Huang, and Robert D. Russell. Comparison of two-dimensional r-adaptive finite element methods using various error indicators. *Mathematics and Computers in Simulation*, 56(2):127 – 143, 2001. Method of lines.

[31] Philip Claude Caplan, Robert Haimes, David L. Darmofal, and Marshall C. Galbraith. Anisotropic geometry-conforming *d*-simplicial meshing via isometric embeddings. In *Proceedings of the 26th International Meshing Roundtable*. Springer Berlin Heidelberg, 2017.

[32] Hugh A. Carson, David L. Darmofal, Marshall C. Galbraith, and Steven R. Allmaras. Analysis of output-based error estimation for finite element methods. *Applied Numerical Mathematics*, 118:182 – 202, 2017.

[33] G. Chavent, M. Dupuy, and P. Lemmonier. History Matching by Use of Optimal Theory. *Society of Petroleum Engineers*, 15(1):74–86, 1975.

[34] Guy Chavent, Gary Cohen, Jerome Jaffre, Robert Eyard, Dominique R. Guerillot, and Luce Weill. Discontinuous and Mixed Finite Elements for Two-Phase Incompressible Flow. *Society of Petroleum Engineers*, 5(4), 1990.

[35] W.H. Chen, G.R. Gavalas, J.H. Seinfeld, and M.L. Wasserman. A New Algorithm for Automatic History Matching. *Society of Petroleum Engineers*, 14(6):593–608, 1974.

[36] Zhiyun Chen, Holger Steeb, and Stefan Diebels. A space-time discontinuous Galerkin method applied to single-phase flow in porous media. *Computational Geosciences*, 12(4):525–539, 2008.

[37] M.A Christie and M.J Blunt. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *Society of Petroleum Engineers*, 2001.

[38] C.C. Chueh, M. Secanell, W. Bangerth, and N. Djilali. Multi-level adaptive simulation of transient two-phase flow in heterogeneous porous media. *Computers & Fluids*, 39(9):1585 – 1596, 2010.

[39] B. Cockburn, S. Hou, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.

[40] B. Cockburn, S. Y. Lin, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.

[41] B. Cockburn and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[42] B. Cockburn and C. W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, December 1998.

[43] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.

[44] Clint Dawson, Shuyu Sun, and Mary F. Wheeler. Compatible algorithms for coupled flow and transport. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2565 – 2580, 2004.

[45] C. Dobrzynski and P. Frey. Anisotropic Delaunay mesh adaptation for unsteady simulations. In *17th Intl. Meshing Roundtable*, pages 177–194, Oct 2008.

[46] Ali H. Dogru. Giga-Cell Simulation. *Saudi Aramco Journal of Technology*, 2011.

[47] Ali H. Dogru, Larry Siu Kuen Fung, Usuf Middya, Tareq Mutlaq Al-Shaalan, Tom Byer, H. Hoy, Werner Artur Hahn, Nabil Al-Zamel, Jorge Alberto Pita, Kesavalu Hemanthkumar, M. Mezghani, Abdulrahman Al-Mana, J. Tan, W. Dreiman, A. Fugl, and Abdulaziz Al-Baiz. New Frontiers in Large Scale Reservoir Simulation. *Society of Petroleum Engineers*, 2011.

[48] Ali H. Dogru, Larry S.K. Fung, Tareq M. Al-Shaalan, Usuf Middya, and Jorge A. Pita. From Mega Cell to Giga Cell Reservoir Simulation. *Society of Petroleum Engineers*, 2008.

[49] Rex A. Dwyer. Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6(3):343–367, 1991.

[50] Michael G. Edwards. A higher-order Godunov scheme coupled with dynamic local grid refinement for flow in a porous medium. *Computer Methods in Applied Mechanics and Engineering*, 131(3):287 – 308, 1996.

[51] Y. Epshteyn and B. Rivière. On the solution of incompressible two-phase flow by a p-version discontinuous Galerkin method. *Communications in Numerical Methods in Engineering*, 22(7):741–751, 2006.

[52] Y. Epshteyn and B. Rivière. Fully implicit discontinuous finite element methods for two-phase flow. *Applied Numerical Mathematics*, 57(4):383 – 401, 2007.

[53] Yekaterina Epshteyn and Béatrice Rivière. Analysis of *hp* discontinuous Galerkin methods for incompressible two-phase flow. *Journal of Computational and Applied Mathematics*, 225(2):487 – 509, 2009.

[54] Dmitry Eydinov, Sigurd Ivar Aanonsen, Jarle Haukås, and Ivar Aavatsmark. A method for automatic history matching of a compositional reservoir simulator with multipoint flux approximation. *Computational Geosciences*, 12(2):209–225, Jun 2008.

[55] Robert Eymard, Thierry Gallouet, and Raphaèle Herbin. Finite volume methods. In *Solution of Equation in $\mathbb{R}^n$ (Part 3), Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713 – 1018. Elsevier, 2000.

[56] Krzysztof Fidkowski and David Darmofal. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA Journal*, 49(4):673–694, 2011.

[57] I. Fried. Finite-element analysis of time-dependent phenomena. *AIAA Journal*, 7(6):1170 – 1173, 1969.

[58] Larry S.-K. Fung, Xiang Y. Ding, and Ali H. Dogru. Unconstrained Voronoi Grids for Densely Spaced Complex Wells in Full-Field Reservoir Simulation. *Society of Petroleum Engineers*, 19(5), 2014.

[59] L.S.-K. Fung, A.D. Hiebert, and L.X. Nghiem. Reservoir Simulation With a Control-Volume Finite-Element Method. *Society of Petroleum Engineers*, 7(3), 1992.

[60] Marshall C. Galbraith, Steven Allmaras, and David L. Darmofal. A Verification Driven Process for Rapid Development of CFD Software. AIAA 2015-0818, January 2015.

[61] M. Giles and N. Pierce. Adjoint error correction for integral outputs. In T.J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick, editors, *Lecture Notes in Computational Science and Engineering: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, volume 25. Springer, Berlin, 2002.

[62] M. B. Giles and E. Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. In *Acta Numerica*, volume 11, pages 145–236, 2002.

[63] M.B. Giles. Discrete adjoint approximations with shocks. In T.Y. Hou and E. Tadmor, editors, *Hyperbolic Problems: Theory, Numerics, Applications*. Springer-Verlag, 2003.

[64] Michael B. Giles and Niles A. Pierce. Adjoint equations in CFD: duality, boundary conditions, and solution behaviour. AIAA 97-1850, 1997.

[65] Michael B. Giles and Niles A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

[66] Michael B. Giles and Niles A. Pierce. Analytic adjoint solutions for the quasi-one dimensional Euler equations. *J. Fluid Mech.*, pages 327–345, 2001.

[67] Mike Giles and Stefan Ulbrich. Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 1: Linearized approximations and linearized output functionals. 48(3):882–904, 2010.

[68] Pedro González-Rodríguez, Manuel Kindelan, Miguel Moscoso, and Oliver Dorn. History matching problem in reservoir engineering using the propagation-backpropagation method. *Inverse Problems*, 21(2):565, 2005.

[69] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230:4248–4267, 2011.

[70] R. Hartmann. Numerical Analysis of Higher Order Discontinuous Galerkin Finite Element Methods. In H. Deconinck, editor, *VKI LS 2008-08: CFD/ADIGMA course on very high order discretization methods*. Von Karman Institute for Fluid Dynamics, 2008.

[71] Ralf Hartmann and Paul Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.

[72] Ralf Hartmann and Paul Houston. Error estimation and adaptive mesh refinement for aerodynamic flows. In H. Deconinck, editor, *VKI LS 2010-01: 36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, Oct. 26-30, 2009*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2009.

[73] A. Hiltebrand and S. Mishra. Entropy stable shock capturing space-time discontinuous Galerkin schemes for systems of conservation laws. *Numerische Mathematik*, 126:103 – 151, 2014.

[74] Hussein Hoteit and Abbas Firoozabadi. Compositional Modeling by the Combined Discontinuous Galerkin and Mixed Methods. *Society of Petroleum Engineers*, 11(1), 2006.

[75] Hussein Hoteit and Abbas Firoozabadi. Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures. *Advances in Water Resources*, 31(1):56 – 73, 2008.

[76] Thomas J. R. Hughes and Gregory M. Hulbert. Space-time finite element methods for elastodynamics: Formulations and error estimates. *Computer Methods in Applied Mechanics and Engineering*, 66:339–363, 1988.

[77] Gregory M. Hulbert and Thomas J. R. Hughes. Space-time Finite Element Methods for Second-order Hyperbolic Equations. *Computer Methods in Applied Mechanics and Engineering*, 84(3):327–348, December 1990.

[78] Savithru Jayasinghe. A Space-time Adaptive Method for Flows in Oil Reservoirs. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, September 2015.

[79] Savithru Jayasinghe, David L. Darmofal, Nicholas K. Burgess, Marshall C. Galbraith, and Steven R. Allmaras. A space-time adaptive method for reservoir flows: formulation and one-dimensional application. *Computational Geosciences*, 22(1):107–123, February 2018.

[80] Savithru Jayasinghe, David L. Darmofal, Marshall C. Galbraith, Nicholas K. Burgess, and Steven R. Allmaras. Adjoint analysis of Buckley-Leverett and two-phase flow equations. *Computational Geosciences*, 22(2):527–542, April 2018.

[81] Guangshan Jiang and Chi-Wang Shu. On a cell entropy inequality for Discontinous Galerkin methods. *Mathematics of Computation*, 62(206):531–538, 1994.

[82] Claes Johnson, Anders Szepessy, and Peter Hansbo. On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws. *Mathematics of Computation*, 54(189):107–129, 1990.

[83] Steven G. Johnson. The NLopt nonlinear-optimization package, available at: http://ab-initio.mit.edu/nlopt.

[84] George Karypis. Parmetis: Parallel graph partitioning and sparse matrix ordering library, 2006. http://glaros.dtc.umn.edu/gkhome/views/metis/parmetis.

[85] W. Klieber and B. Rivière. Adaptive simulations of two-phase flow by discontinuous galerkin methods. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):404 – 419, 2006.

[86] Drosos Kourounis, Louis J. Durlofsky, Jan Dirk Jansen, and Khalid Aziz. Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Computational Geosciences*, 18(2):117–137, Apr 2014.

[87] J. Kudo. Robust adaptive high-order RANS methods. Master's thesis, Mass. Inst. of Tech., CDO, June 2014.

[88] Sanghyun Lee and Mary F. Wheeler. Adaptive enriched Galerkin methods for miscible displacement problems with entropy residual stabilization. *Journal of Computational Physics*, 331:19 – 37, 2017.

[89] S.H. Lee and W.J. Milliken. The Productivity Index of an Inclined Well in Finite-Difference Reservoir Simulation. In *SPE Symposium on Reservoir Simulation*. Society of Petroleum Engineers, 1993.

[90] John Leech. Some sphere packings in higher space. *Canadian Journal of Mathematics*, pages 657–682, 1964.

[91] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Lectures in Mathematics ETH Zürich, Department of Mathematics Research Institute of Mathematics. Birkhäuser Basel, 1992.

[92] Jizhou Li and Beatrice Riviere. High order discontinuous Galerkin method for simulating miscible flooding in porous media. *Computational Geosciences*, 19(6):1251–1268, Dec 2015.

[93] Ruijian Li, A.C. Reynolds, and D.S. Oliver. History Matching of Three-Phase Flow Production Data. *Society of Petroleum Engineers*, 2003.

[94] Hui Liu, Kun Wang, Zhangxin Chen, and Kirk E. Jordan. Efficient Multi-stage Preconditioners for Highly Heterogeneous Reservoir Simulations on Parallel Distributed Systems. *Society of Petroleum Engineers*, 2015.

[95] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3d steady Euler equations. *Journal of Computational Physics*, 229:2866–2897, 2010.

[96] Adrien Loseille and Frédéric Alauzet. Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework. In *Proceedings of the 18th International Meshing Roundtable*, pages 575–594. Springer Berlin Heidelberg, 2009.

[97] Adrien Loseille and Rainald Löhner. On 3d anisotropic local remeshing for surface, volume and boundary layers. In *Proceedings of the 18th International Meshing Roundtable*, pages 611–630. Springer Berlin Heidelberg, 2009.

[98] Adrien Loseille and Rainald Löhner. Anisotropic adaptive simulations in aerodynamics. AIAA 2010-169, 2010.

[99] James M. Modisette. *An Automated Reliable Method for Two-Dimensional Reynolds-averaged Navier-Stokes Simulations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, September 2011.

[100] Joachim Moortgat and Abbas Firoozabadi. Higher-order compositional modeling of three-phase flow in 3d fractured porous media based on cross-flow equilibrium. *Journal of Computational Physics*, 250:425 – 445, 2013.

[101] David Moro, N.C. Nguyen, and J. Peraire. Navier-Stokes solutions using hybridizable discontinuous Galerkin methods. AIAA 2011-3407, 2011.

[102] D. Nayagum, G. Schäfer, and R. Mosé. Modelling two-phase incompressible flow in porous media using mixed hybrid and discontinuous finite elements. *Computational Geosciences*, 8(1):49–73, Mar 2004.

[103] Martin Neumüller and Olaf Steinbach. Refinement of flexible space-time finite element meshes and discontinuous galerkin methods. *Comput Visual Sci*, 14:189–205, 2011.

[104] E. Obi, N. Eberle, A. Fil, and H Cao. Giga Cell Compositional Simulation. *International Petroleum Technology Conference*, 2014.

[105] J. T. Oden, I. Babuska, and C. E. Baumann. A discontinuous *hp* finite element method for diffusion problems. *Journal of Computational Physics*, 146:491–519, 1998.

[106] J. Tinsley Oden. A general theory of finite elements. II. applications. *International Journal for Numerical Methods in Engineering*, 1(3):247–259, 1969.

[107] D. S. Oliver, A. C. Reynolds, and N. Liu. *Inverse Theory for Petroleum Reservoir Characterization and History Matching.* 2008.

[108] Todd Oliver and David Darmofal. Analysis of dual consistency for discontinuous Galerkin discretizations of source terms. *SIAM Journal on Numerical Analysis*, 47(5):3507–3525, 2009.

[109] Todd A. Oliver. *A Higher-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-averaged Navier-Stokes Equations.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2008.

[110] C.L. Palagi and Khalid Aziz. Modeling vertical and horizontal wells with voronoi grid. *Society of Petroleum Engineers*, 9(1), 1994.

[111] George Shu Heng Pau, John B. Bell, Ann S. Almgren, Kirsten M. Fagnan, and Michael J. Lijewski. An adaptive mesh refinement algorithm for compressible two-phase flow in porous media. *Computational Geosciences*, 16(3):577–592, Jun 2012.

[112] D.W Peaceman. *Fundamentals of Numerical Reservoir Simulation.* Elsevier, 1977.

[113] D.W. Peaceman. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation. *Society of Petroleum Engineers*, 18(3), 1978.

[114] D.W. Peaceman. Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability. *Society of Petroleum Engineers*, 23(3), 1983.

[115] D.W. Peaceman. Interpretation of Wellblock Pressures in Numerical Reservoir Simulation: Part 3 – Off-Center and Multiple Wells Within a Wellblock. *Society of Petroleum Engineers*, 5(2), 1990.

[116] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[117] Jaime Peraire and Per-Olof Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[118] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. AIAA 2006-0112, 2006.

[119] Per-Olof Persson and Jaime Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2722, 2008.

[120] Richard Rankin and Beatrice Riviere. A high order method for solving the black-oil problem in porous media. *Advances in Water Resources*, 78:126 – 144, 2015.

[121] Béatrice Rivière. Numerical study of a discontinuous Galerkin method for incompressible two-phase flow. In *ECCOMAS Proceedings*, Finland, 2004.

[122] Béatrice Rivière and Mary F. Wheeler. Discontinuous Galerkin methods for flow and transport problems in porous media. *Communications in Numerical Methods in Engineering*, 18(1):63–68, 2002.

[123] Béatrice Rivière, Mary F. Wheeler, and Krzysztof Banaś. Part II. Discontinuous Galerkin method applied to a single phase flow in porous media. *Computational Geosciences*, 4:337–349, 2000.

[124] Béatrice Rivière, Mary F. Wheeler, and Vivette Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. Part I. *Computational Geosciences*, 3(3-4):337–360, 1999.

[125] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.

[126] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[127] Paula A. Sesini and Alvaro L. G. A. Coutinho. Stabilized Finite Element Methods for Three-phase Porous Media Flows. *MecÃ̧nica Computacional*, XXIX:8753–8765, 2010.

[128] Paula A. Sesini, Denis A. F. de Souza, and Alvaro L. G. A. Coutinho. Finite element simulation of viscous fingering in miscible displacements at high mobility-ratios. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 32:292 – 299, 09 2010.

[129] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.

[130] D. A. Venditti and D. L. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *Journal of Computational Physics*, 176(1):40–69, 2002.

[131] D. A. Venditti and D. L. Darmofal. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *Journal of Computational Physics*, 187(1):22–46, 2003.

[132] J.R. Wallis, R.P. Kendall, and T.E. Little. Constrained Residual Acceleration of Conjugate Residual Methods. *Society of Petroleum Engineers*, 1985.

[133] Z.J. Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, H.T. Huynh, Norbert Kroll, Georg May, Per-Olof Persson, Bram van Leer, and Miguel Visbal. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[134] M.L. Wasserman, A.S. Emanuel, and J.H. Seinfeld. Practical Applications of Optimal-Control Theory to History-Matching Multiphase Simulator Models. *Society of Petroleum Engineers*, 15(4):347–355, 1975.

[135] M. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15:152–161, 1978.

[136] Zhan Wu, A.C. Reynolds, and D.S. Oliver. Conditioning Geostatistical Models to Two-Phase Production Data. *Society of Petroleum Engineers*, 1999.

[137] Masayuki Yano. *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes.* PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, June 2012.

[138] Masayuki Yano and David L. Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, September 2012.

[139] M. Zakerzadeh and G. May. On the convergence of a shock capturing discontinuous Galerkin method for nonlinear hyperbolic systems of conservation laws. *SIAM Journal on Numerical Analysis*, 54(2), March 2016.

[140] Chen Zhangxin and Zhang Youqian. Well flow models for various numerical methods. *International Journal of Numerical Analysis & Modeling*, 6(3):375 – 388, 2009.

[141] Valentin Zingan, Jean-Luc Guermond, Jim Morel, and Bojan Popov. Implementation of the entropy viscosity method with the discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 253:479 – 490, 2013.

[142] C. Zong. *Sphere Packings.* Springer New York, 1999.