A Space-time Adaptive Method for Flows in Oil Reservoirs

by

Yashod Savithru Jayasinghe

M.Eng., University of Cambridge (2013)

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

© Massachusetts Institute of Technology 2015. All rights reserved.
A Space-time Adaptive Method for Flows
in Oil Reservoirs

by

Yashod Savithru Jayasinghe

Submitted to the Department of Aeronautics and Astronautics
on August 20, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This work presents a space-time adaptive framework for simulating multi-phase flows through porous media, with specific applications to flows in oil reservoirs. A fully unstructured discretization of space and time is used instead of a conventional time-marching approach. For $d$-dimensional spatial problems, this requires the generation of $(d+1)$-dimensional meshes, where time is treated as an additional spatial dimension. Anisotropic mesh adaptation is performed based on a posteriori error estimation to reduce the error of a specified output of interest. This work makes use of the DWR method for error estimation and the MOESS algorithm for metric-based mesh optimization. A discontinuous Galerkin finite element discretization is used to solve on simplex meshes with arbitrary anisotropy, and thereby obtain solutions of higher order accuracy in both space and time. The adaptive framework has been applied to single-phase and two-phase flow test problems in a one-dimensional reservoir, and the results were compared to those obtained from a time-marching finite volume method that is representative of a typical industrial simulator.

Thesis Supervisor: David Darmofal
Title: Professor of Aeronautics and Astronautics
Acknowledgments

I would like to express my gratitude to all those who have helped in numerous ways to make this research study and thesis possible, while making my time here at MIT a memorable one.

Firstly, I would like to thank my advisor, Prof. David Darmofal, for giving me the opportunity to be a part of this wonderful research group, and for his consistent guidance and encouragement over these past two years. I definitely look forward to continue working together. I would also like to thank Dr. Steven Allmaras for his valuable insights and advice throughout this project. Marshall, thank you for all the help in implementing (read: debugging) code, for keeping the ProjectX code sane, and for the feedback on this thesis. I would also like to thank Nick for his valuable input and support in shaping the course of this project.

This work would have not been possible without the efforts and contributions of the entire ProjectX team, both past and present. I would specifically like to thank Jun, Phil, and Steve for helping us settle down in the lab, and Masa for laying down the foundations of the adaptive framework that is used in this work. Yixuan, Carlee, Jeff, Arthur, and Hugh, thank you for all the useful discussions and support throughout, not to mention the countless hours spent doing psets together. I wish the best of luck to all of you.

I would also like to thank everyone in the ACDL for creating a productive and friendly working environment. In particular, I would like to thank Eric and Phil for keeping the computational resources in the lab functioning smoothly, and also for helping me, Chaitanya, and Chi transition into our new roles as system administrators. Special thanks to Patrick, Rémi, Alessio, Chai, Yixuan, Zheng, Harriet, and Angxiu for the support in preparing for quals last January. I’m also grateful to all my Sri Lankan friends here for the eventful weekends, Reetik for the much needed badminton-breaks, and the entire MIT cricket team for the wonderful experiences. Thank you Thanuja for the random Skype calls from the other end of the world, and for being a great friend all these years. Narmada, I’m sincerely thankful for all the
love, support, and wonderful memories over the past two years, and for making MIT feel home.

Lastly, I would like to thank my parents and sister for their unconditional love and constant encouragement, without which I would not have gotten this far. 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. Nick Burgess.
## Contents

1 Introduction  
  1.1 Motivation .................................................. 15  
  1.2 Background .................................................. 16  
    1.2.1 High-order methods .................................. 16  
    1.2.2 Solution adaptive methods ......................... 18  
    1.2.3 Space-time adaptive methods ...................... 21  
  1.3 Thesis overview ........................................... 23  

2 Discretization, Error Estimation, and Output-based Adaptation  
  2.1 Discretization ............................................... 27  
    2.1.1 Inviscid discretization .............................. 30  
    2.1.2 Viscous discretization .............................. 31  
    2.1.3 Source discretization ............................... 32  
  2.2 Solution method ........................................... 32  
    2.2.1 Line search .......................................... 33  
    2.2.2 Continuation methods ............................... 34  
  2.3 Output error estimation .................................. 36  
    2.3.1 Dual-weighted residual method .................... 37  
    2.3.2 Error localization .................................. 38  
  2.4 Mesh adaptation ........................................... 39  
    2.4.1 Continuous mesh framework ....................... 39  
    2.4.2 Mesh Optimization via Error Sampling and Synthesis .... 40
### 3 Basic Concepts and Definitions

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Representative elementary volume</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>Porosity</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Saturation</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Darcy’s law and permeability</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Capillary pressure</td>
<td>49</td>
</tr>
<tr>
<td>3.6</td>
<td>Compressibility</td>
<td>50</td>
</tr>
</tbody>
</table>

### 4 Single-Phase Flow

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Governing equations</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Space-time formulation</td>
<td>55</td>
</tr>
<tr>
<td>4.3</td>
<td>Test problem</td>
<td>56</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Problem statement</td>
<td>56</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Objectives</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Numerical results</td>
<td>59</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Adapted solutions</td>
<td>59</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Comparisons with a conventional method</td>
<td>61</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Other observations</td>
<td>66</td>
</tr>
</tbody>
</table>

### 5 Buckley-Leverett Equation

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Background</td>
<td>75</td>
</tr>
<tr>
<td>5.2</td>
<td>Governing equations</td>
<td>76</td>
</tr>
<tr>
<td>5.3</td>
<td>Space-time formulation</td>
<td>77</td>
</tr>
<tr>
<td>5.4</td>
<td>Test problem</td>
<td>77</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Problem statement</td>
<td>77</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Objectives</td>
<td>78</td>
</tr>
<tr>
<td>5.5</td>
<td>Numerical results</td>
<td>81</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Hyperbolic equation</td>
<td>81</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Nearly-hyperbolic equation</td>
<td>88</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Comparisons with a conventional method</td>
<td>92</td>
</tr>
<tr>
<td>5.6</td>
<td>Summary</td>
<td>94</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>General outline of adaptation framework</td>
</tr>
<tr>
<td>1-2</td>
<td>Illustration of different space-time meshes (Yano [56])</td>
</tr>
<tr>
<td>2-1</td>
<td>Mesh metric-field duality (Modisette [39])</td>
</tr>
<tr>
<td>2-2</td>
<td>Example split configurations with associated metric tensors (Yano [55])</td>
</tr>
<tr>
<td>4-1</td>
<td>Schematic of 1D reservoir</td>
</tr>
<tr>
<td>4-2</td>
<td>Schematic of space-time domain</td>
</tr>
<tr>
<td>4-3</td>
<td>Adapted space-time primal solution - P2 50k DOF</td>
</tr>
<tr>
<td>4-4</td>
<td>Adapted space-time mesh - P2 50k DOF</td>
</tr>
<tr>
<td>4-5</td>
<td>Adapted space-time adjoint solution - P2 50k DOF</td>
</tr>
<tr>
<td>4-6</td>
<td>Pressure history at ( x = 1000 ) ft</td>
</tr>
<tr>
<td>4-7</td>
<td>Zoomed-in view of pressure history at ( x = 1000 ) ft</td>
</tr>
<tr>
<td>4-8</td>
<td>Error vs. ( h )</td>
</tr>
<tr>
<td>4-9</td>
<td>Error vs. Runtime</td>
</tr>
<tr>
<td>4-10</td>
<td>Adapted space-time mesh - P1 50k DOF</td>
</tr>
<tr>
<td>4-11</td>
<td>Adapted space-time mesh - P2 50k DOF</td>
</tr>
<tr>
<td>4-12</td>
<td>Second spatial derivative of adapted primal solution - ( \left</td>
</tr>
<tr>
<td>4-13</td>
<td>Second time derivative of adapted primal solution - ( \left</td>
</tr>
<tr>
<td>4-14</td>
<td>Second cross derivative of adapted primal solution - ( \left</td>
</tr>
<tr>
<td>4-15</td>
<td>Zoomed-in view of ( \left</td>
</tr>
<tr>
<td>4-16</td>
<td>Zoomed-in view of ( \left</td>
</tr>
<tr>
<td>4-17</td>
<td>Zoomed-in view of ( \left</td>
</tr>
<tr>
<td>4-18</td>
<td>Second spatial derivative of adapted adjoint solution - ( \left</td>
</tr>
</tbody>
</table>
4-19 Second time derivative of adapted adjoint solution - $\left| \frac{\partial^2 \psi}{\partial t^2} \right| \times T^2$ . . . . 72
4-20 Second cross derivative of adapted adjoint solution - $\left| \frac{\partial^2 \psi}{\partial x \partial t} \right| \times LT$ . . . . 73
4-21 Zoomed-in view of $\left| \frac{\partial^2 \psi}{\partial x^2} \right| \times L^2$ with overlaid mesh . . . . . . . . . . . 73
4-22 Zoomed-in view of $\left| \frac{\partial^2 \psi}{\partial x \partial t} \right| \times T^2$ with overlaid mesh . . . . . . . . . . . 74
4-23 Zoomed-in view of $\left| \frac{\partial^2 \psi}{\partial x^2} \right| \times LT$ with overlaid mesh . . . . . . . . . . . 74

5-1 Functions for the Buckley-Leverett problem . . . . . . . . . . . . . . . . 79
5-2 Saturation at $t = 30$ days, with quadratic $k_r$ . . . . . . . . . . . . . . 82
5-3 Saturation at $t = 30$ days, with quartic $k_r$ . . . . . . . . . . . . . . 82
5-4 Space-time mesh adapted to $J_1$ - P0 10k DOF with quadratic $k_r$ . . . 83
5-5 Space-time mesh adapted to $J_2$ - P0 10k DOF with quadratic $k_r$ . . . 83
5-6 Space-time mesh adapted to $J_1$ - P0 10k DOF with quartic $k_r$ . . . . 84
5-7 Space-time mesh adapted to $J_2$ - P0 10k DOF with quartic $k_r$ . . . . 84
5-8 Adjoint solution adapted to $J_1$ - P0 10k DOF with quadratic $k_r$ . . . 85
5-9 Adjoint solution adapted to $J_2$ - P0 10k DOF with quadratic $k_r$ . . . 85
5-10 Saturation at $t = 30$ days, with quadratic $k_r$ - P1 solutions . . . . . . 87
5-11 Space-time solution - P1 50x60 with quadratic $k_r$ . . . . . . . . . . . 87
5-12 Adapted space-time primal solution - P1 30k . . . . . . . . . . . . . . 90
5-13 Adapted space-time adjoint solution - P1 30k . . . . . . . . . . . . . . 90
5-14 Adapted space-time mesh - P1 30k . . . . . . . . . . . . . . . . . . . . 91
5-15 Adapted space-time mesh - P2 30k . . . . . . . . . . . . . . . . . . . . 91
5-16 Comparison of $S_w$ profiles at $t = 30$ days . . . . . . . . . . . . . . 92
5-17 Error vs. $h$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
5-18 Error vs. Runtime . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95

6-1 Schematic of 1D reservoir . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
6-2 Adapted space-time solution - P2 30k DOF - pressure ($p_w$) . . . . . . 106
6-3 Adapted space-time solution - P2 30k DOF - saturation ($S_w$) . . . . . 106
6-4 Adapted space-time solution - P2 30k DOF - adjoint $\psi_w$ . . . . . . 107
6-5 Adapted space-time solution - P2 30k DOF - adjoint $\psi_n$ . . . . . . 107
6-6 Adapted space-time mesh - P1 30k DOF - full domain . . . . . . . . . . 108
Chapter 1

Introduction

Numerical simulation has become an important tool for understanding and predicting the performance of reservoirs. In the context of hydrocarbon reservoirs, numerical simulations are used to investigate flow processes, assess the viability of recovery methods, and predict the overall reservoir performance under different operating conditions. Since these simulation results eventually have a significant impact on the engineering and management decisions that are made, the accuracy of these results is of great importance.

1.1 Motivation

A computational fluid dynamics (CFD) model 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. A common way of improving solution fidelity is to increase the mesh resolution by adding more elements, but this cannot be done indefinitely due to limitations in computing power. Even with recent advances in parallel computing, most large scale reservoir simulators in use today can solve problems with hundreds of millions of cells, with the most powerful simulators only just 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, and all heterogeneous features within that block are averaged out [21]. However, the seismic data obtained from geological surveys are usually of much greater resolution, showing that CFD models still have room for improvement.

Continued efforts at increasing the fidelity of the CFD model by adding more elements into the mesh have shown significant improvements in reservoir performance predictions [22, 23]. Finer meshes allow the model to accurately capture features such as sharp saturation fronts, gas breakthroughs, and regions of trapped oil, all of which affect the performance of the reservoir. However, this approach is not only greatly limited by computational resources, but also needs to be done carefully to be productive. 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 are often riddled with prominent solution features that need to be captured accurately. However, knowing the size, location and orientation of these features beforehand is a non-trivial task, except maybe in the case of geological heterogeneities. In such cases where the optimal mesh cannot be found a priori, a more attractive alternative is to develop an adaptive method that can autonomously modify the mesh, or more generally the discretization, according to the solution to produce a more reliable and accurate output.

1.2 Background

1.2.1 High-order methods

Flows in reservoirs are generally governed by a variety of physical phenomena, which causes the governing mathematical equations to be highly complex and nonlinear in nature. For the past few decades, much of the reservoir simulation community and industry has used low-order discretizations based on the finite volume method (FVM)
and the finite difference method (FDM) to model reservoir flows. The reliability and robustness of these methods has allowed the industry to successfully solve large and complex reservoir flow problems over the years. The term “low-order” is typically used to identify numerical methods which have at most second-order accuracy in space and time. A numerical method is said to be of order \( r \) if some measure of the solution error \( e \) (i.e. \( L^2 \) error) is proportional to the mesh size \( h \) to the power \( r \), i.e. \( e \propto \mathcal{O}(h^r) \).

As with any numerical simulation, the results of reservoir simulation are also susceptible to various uncertainties and errors. These errors can be categorized into two primary components, modeling errors and numerical errors. For example, errors that arise due to the inaccuracy of the geological model, which represents the physical properties of the reservoir, can be identified as modeling errors. On the other hand, even if all physical properties are exactly known and all physical phenomena are accurately captured by the governing equations, the process of numerical discretization still introduces errors into the solution. These are referred to as numerical errors. Note that numerical errors may also include errors caused by the finite precision arithmetic of computers, but these are assumed to be negligible compared to the other sources of error. With increasingly accurate seismic surveys and the use of modern seismic inversion methods, geologists have been able to reduce the magnitude of modeling errors, but unless numerical errors are also reduced in a similar fashion, the overall accuracy of the solution will not improve significantly. The common practice for reducing numerical errors involves refining the mesh and reducing the size of the timestep for unsteady problems. However, doing this with low-order numerical methods soon causes the size of the problem to become computationally infeasible.

In recent years, high-order methods have been increasingly popular for solving porous media flow problems. Finite element methods offer a rather straightforward way to obtain high-order accurate solutions by increasing the order of the polynomial basis functions. In particular, schemes such as the discontinuous Galerkin finite element method (DG) have been successfully applied to single phase, multi-phase and linear transport flow problems. One of the main advantages of this
method is its local nature, where unlike for finite volume or finite difference methods, the size of the element stencil remains small even for higher orders of approximation. Together with other useful properties such as local mass conservation, ease of implementation on unstructured grids, and general robustness for capturing discontinuous solutions, the DG method offers a competitive alternative to the conventional low-order methods. The elementwise discontinuous polynomial representation of the DG method also makes it a favorable scheme for performing $hp$-adaptation.

Due to the higher convergence rates, high-order methods generally require fewer degrees of freedom to achieve the same level of accuracy as low-order methods, for smooth problems. Although high-order methods may be more expensive than low-order methods on a given mesh, the trade-off between the number of degrees of freedom required usually enables high-order methods to be more efficient, in terms of overall computational cost. The efficiency gains of increasing the polynomial order ($p$-adaptation) have been observed to be especially true for problems with smooth solutions [5]. However for problems with low regularity, as is generally the case for advection dominated flows, the benefits of high-order methods may not be realized without also utilizing mesh adaptation ($h$-adaptation).

1.2.2 Solution adaptive methods

The objective of a numerical reservoir simulation is to accurately predict outputs of interest, which can then be used to make important engineering or management decisions. For example, the total amount of oil produced from an oil reservoir over a given time period can be a useful output from a numerical simulation. The accuracy of the output greatly depends on the mesh that is used in the numerical simulation, and hence it is common for the mesh generation process to involve numerous “best practices”, which are based on previous experience and insights into the specific problem.

In contrast, a solution adaptive method can autonomously arrive at accurate estimates of outputs without any prior knowledge of the problem. It is also worth noting that most of the best practice techniques involved in mesh generation are
based on the experience gained from using low-order methods. For example, an ef-
ficient mesh for a third-order method may be significantly different from that for a
second-order method, and recognizing those differences may prove to be a non-trivial
task for a human. A solution adaptive framework minimizes the amount of human
intervention and therefore can systematically produce a reliable output. In this work,
this is achieved via *a posteriori* output-based error estimation and mesh adaptation
algorithms.

**Outline**

The general outline of the output-based solution adaptation framework can be de-
scribed using Figure 1-1 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 to de-
ote note the amount of computational resources available (i.e. 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, the adap-
tation 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 time.

![Figure 1-1: General outline of adaptation framework](image-url)
Error estimation

The error estimation algorithm forms a key component of the adaptation framework since it identifies the regions in the mesh where the contribution to the output error is large. There exist a few varieties of error estimation strategies. Tracking the magnitude of gradients to identify interesting solution features is a commonly used method [6, 14, 17], but it should be noted that the presence of large solution gradients does not necessarily imply that the solution errors are large in that region. Residual-based error estimation is another variant, as demonstrated for porous media flows by Klieber using the DG method in [33] and by Amaziane et al. using FVM in [1]. However for advection dominant flows, it is known that upstream errors can propagate and pollute the solution downstream. Hence, there exist error estimation techniques that incorporate the adjoint solution from the dual problem, which captures the sensitivity of the output of interest to perturbations in the primal residual. This work uses the dual-weighted residual (DWR) approach proposed by Becker and Rannacher [10, 11] to obtain global and local error estimates, which are then used to drive the mesh adaptation.

Adaptation

Once the error estimates have been computed on the current mesh, the next step of the adaptation process is to modify the current discretization to decrease the error estimates further. For low-order methods, this is almost always done by reducing the grid spacing locally in regions where the errors are large. However, modifications to the discretization can be categorized broadly into three types: $h$-adaptation, $p$-adaptation and $hp$-adaptation.

$h$-adaptation involves the modification of the mesh, and is probably the most common choice. In this method, the size and shape of the elements in the mesh are modified in order to reduce the overall error in the output. A widely used strategy is to perform isotropic mesh refinement where selected elements are uniformly refined to decrease the error, as seen in [33, 1, 17] for flows through heterogeneous porous media.
Recent work has also shown successful demonstrations of anisotropic $h$-adaptation. Venditti and Darmofal [52] have shown output-based anisotropic adaptation results for the Navier-Stokes equations based on the Hessian of the Mach number. More recently, the Mesh Optimization via Error Sampling and Synthesis (MOESS) algorithm proposed by Yano and Darmofal [55] performs anisotropic adaptation by constructing surrogate error models via element-wise local solves and then using this error model to obtain an optimal metric field that minimizes the output error for a given computational cost. A metric tensor field is used to represent the size and orientation of each element in the mesh.

In $p$-adaptation, the discretization is modified by changing the order of accuracy locally, while keeping the mesh fixed. In the context of finite element methods, this can be done by changing the order of the polynomial basis functions in selected elements. Although $p$-adaptation has been observed to be more efficient for smooth problems, they have not been very effective for problems with low regularity, such as those with shocks and other discontinuous features.

$hp$-adaptation promises the best of both worlds, by potentially coupling both $h$ and $p$ adaptation. However, the manner in which this coupling needs to be done is non-trivial and hence this method is still very much an active research topic.

The work in this thesis focuses only on $h$-adaptation, for a given order of the polynomial basis. In particular, Yano’s MOESS framework will be used to perform mesh adaptation, together with Kudo’s modifications to MOESS’s metric optimization algorithm [34].

### 1.2.3 Space-time adaptive methods

Reservoir flow problems are unsteady and typically involve a wave-like feature that propagates through the spatial domain, such as saturation fronts in the case of multiphase flows and viscous fingers in miscible flow problems. In order to resolve these features accurately, the temporal accuracy of the numerical simulation is as important as its spatial accuracy.

Most reservoir simulations utilize first or second order approximations in time,
with the Backward Euler (BDF1) method being a popular choice [4, 44, 48]. The standard approach is to discretize the partial differential equations in space using FVM, FDM, or even FEM, to produce a set of ordinary differential equations which can then be solved with an explicit or implicit time-stepping scheme. However in the case of finite element discretizations, an appealing alternative would be to extend the finite element method along the time axis as well. The idea of using this “space-time finite element method” dates back to the late 1960s, to the work of Oden [41], Argyris and Scharpf [2], and Fried [27].

Benefits of the space-time finite element method include the ability to straightforwardly increase the temporal accuracy by increasing the polynomial order of the space-time element. Furthermore, space-time methods have the potential to break free of the “time-slab” or tensor-product mesh restriction present in all classical time-marching techniques. This allows the method to exploit one of the most useful features of finite element methods: unstructured meshes. The use of a fully unstructured space-time mesh provides a more convenient and natural alternative to the concept of local time-stepping used in time-marching techniques, where different parts of the spatial domain are evolved with different time-steps. The work of Hughes and Hulbert in [30] and [31] present the first use of a discontinuous Galerkin approach in time, which was used to solve second-order hyperbolic problems with applications in elastodynamics. More recently, Chen et al. [16] have developed a DG method which is discontinuous in both space and time to solve a single-phase porous media flow problem. However, they remain within the concept of decoupled time-slabs, where the final value of one time-slab serves as the initial condition for the next.

In [56], Yano and Darmofal demonstrate that using a space-time DG method with fully-unstructured anisotropic space-time mesh adaptation can significantly improve the error-to-degrees-of-freedom efficiency for solving wave propagation problems, compared to uniform mesh refinement or tensor-product space-time adaptation. They motivate their method by comparing the number of space-time degrees of freedom required to accurately capture an important flow feature of characteristic length $\delta \ll L$, with different types of space-time meshes, where $L$ is the domain length. Assuming
the flow feature is transported under the 1D linear advection equation, 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 advection-dominant problems.

![Space-time meshes](image)

Figure 1-2: Illustration of different space-time meshes (Yano [56])

They successfully combine space-time DG formulations for the wave equation and Euler equations, with the MOESS [55] mesh adaptation framework described in the previous subsection. The work of this thesis draws significant inspiration from the aforementioned paper, and aims to apply the same strategies for reservoir flow problems.

### 1.3 Thesis overview

The broad objective of this work is to develop a numerical method for solving reservoir flow problems that can autonomously adapt with the solution to compute specific outputs of interest as accurately as possible for a given computational cost.

Towards this end, our work attempts to unify three distinct approaches which were described in the previous subsection: high-order methods, solution adaptive methods, and space-time adaptive methods. In particular, this thesis makes the
following contributions:

- Formulates a space-time discontinuous Galerkin method for (1+1)d single-phase and two-phase porous media flow problems.

- Demonstrates the solution adaptive framework with higher order discretizations of the (1+1)d single and two-phase flow equations.

- Compares the accuracy and efficiency of the space-time adaptive method with other conventional methods.

Chapter 2 presents the space-time DG method and reviews the DWR method for output error estimation, followed by a summary of the MOESS mesh adaptation framework.

Chapter 3 introduces some of the basic concepts used in porous media flows and provides brief definitions of the key parameters used in this thesis.

Chapter 4 presents a space-time formulation for the single phase “pressure” equation, and applies the solution adaptive framework to a single phase problem in a 1D spatial reservoir. The final adapted results are presented and compared with those from a conventional time-marching finite volume method.

Chapter 5 involves a more general form of the Buckley-Leverett equation that includes capillary effects, and demonstrates the application of the space-time adaptive framework to a test problem. It also includes a discussion of the difficulties of solving the purely hyperbolic Buckley-Leverett equation using high-order methods, and a way of mitigating them. A brief comparison of the results with those from a conventional FV method is also provided.

Chapter 6 presents a space-time formulation for the compressible two-phase flow equations in mass conservation form. The space-time adaptive framework is demonstrated on a test problem, followed by a comparison of the adaptive results with those obtained on uniformly refined structured meshes. Finally, a few limitations of the proposed method are discussed together with some incomplete adaptive results for a highly nonlinear problem.
Chapter 7 contains a summary of the work presented in this thesis and a discussion of how it can be extended and improved in the future.
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 [10, 11] is presented as way of estimating the output error. Finally, a summary of the MOESS framework for mesh adaptation presented by Yano and Darmofal [55] is given.

2.1 Discretization

Let $\Omega_s \in \mathbb{R}^d$ be an arbitrary, bounded domain in a $d$-dimensional space, and $I \in \mathbb{R}^+$ be the time interval of interest. Any $d$-dimensional unsteady conservation law of the form:

$$\frac{\partial u}{\partial t} + \sum_{j=1}^{d} \frac{\partial}{\partial x_j} F_{j}^{\text{inv}}(u, x, t) - \sum_{j=1}^{d} \frac{\partial}{\partial x_j} F_{j}^{\text{visc}}(u, \nabla u, x, t) = S(u, \nabla u, x, t), \quad \forall x \in \Omega_s, \ t \in I,$$

(2.1)
can be recast into a \((d + 1)\)-dimensional “steady-state” conservation law as follows:

\[
\sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} \mathcal{F}^{\text{inv}}_j(u, \hat{x}) - \sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} \mathcal{F}^{\text{visc}}_j(u, \hat{\nabla} u, \hat{x}) = S(u, \hat{\nabla} u, \hat{x}), \quad \forall \hat{x} \in \Omega, \tag{2.2}
\]

where \(\Omega = \Omega_s \cup I \in \mathbb{R}^{d+1}\) is the space-time domain, \(\hat{x} = [x, t] \in \mathbb{R}^{d+1}\) is the augmented space-time coordinate, and \(u(\hat{x}) \in \mathbb{R}^m\) is the \(m\)-variable state vector. The space-time inviscid flux \(\hat{\mathcal{F}}^{\text{inv}}(u, \hat{x}) \in \mathbb{R}^{m \times (d+1)}\), and the space-time viscous flux \(\hat{\mathcal{F}}^{\text{visc}}(u, \hat{\nabla} u, \hat{x}) \in \mathbb{R}^{m \times (d+1)}\) can be written in terms of the spatial fluxes in Eq. (2.1) as:

\[
\hat{\mathcal{F}}^{\text{inv}}(u, \hat{x}) = \begin{bmatrix} \mathcal{F}^{\text{inv}}, u \end{bmatrix} \tag{2.3}
\]

\[
\hat{\mathcal{F}}^{\text{visc}}(u, \hat{\nabla} u, \hat{x}) = \begin{bmatrix} \mathcal{F}^{\text{visc}}, 0 \end{bmatrix} \tag{2.4}
\]

The viscous flux is also assumed to be a linear function of \(\hat{\nabla} u\), and is expressed as:

\[
\hat{\mathcal{F}}^{\text{visc}}(u, \hat{\nabla} u, \hat{x}) = \hat{A}(u, \hat{x}) \hat{\nabla} u \tag{2.5}
\]

\(S(u, \hat{\nabla} u, \hat{x}) \in \mathbb{R}^m\) is the source term and \(\mathcal{B}\) imposes the boundary conditions:

\[
\mathcal{B}(u, \hat{\mathcal{F}}^{\text{visc}}(u, \hat{\nabla} u, \hat{x}) \cdot \hat{n}, \hat{x}; BC) = 0, \quad \forall \hat{x} \in \partial \Omega \tag{2.6}
\]

The initial condition of the original unsteady conservation law is transformed by the above formulation into a Dirichlet boundary condition along the “\(t = 0\)" boundary of the space-time domain \(\Omega\). This “temporal” boundary condition is implemented like any other spatial boundary condition, using \(\mathcal{B}\) as given in Eq. 2.6.

Note that in Eq. (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. For the rest of this chapter, a space-time formulation is always assumed, and therefore the hat accents will be omitted for clarity.

The unsteady governing equations of the reservoir flow problems considered in this thesis will naturally fit into the above formulation. The details of how this is done will be elaborated with the specific equations in later chapters. The rest of this
chapter will continue to assume that the governing equations have already been recast into the form of Eq. 2.2.

The discontinuous Galerkin discretization seeks a solution in a finite dimensional function space $V_{h,p}$, which is defined as:

$$V_{h,p} \equiv \left\{ v \in [L^2(\Omega)]^m : v|_\kappa \in [P^p(\kappa)]^m, \forall \kappa \in T_h \right\}, \quad (2.7)$$

and represents the piecewise discontinuous solution space of $p^{th}$-order polynomials on each element of $T_h$, where $T_h$ is a triangulation of the space-time domain $\Omega$ into non-overlapping elements $\kappa$ of characteristic size $h$.

Multiplying Eq. 2.2 by a test function $v_{h,p} \in V_{h,p}$ and integrating by parts yields the weak formulation of the governing equation. Solving this weak formulation involves finding a solution $u_{h,p} \in V_{h,p}$ that satisfies:

$$\mathcal{R}_{h,p}(u_{h,p}, v_{h,p}) = 0 \quad \forall v_{h,p} \in V_{h,p}, \quad (2.8)$$

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

$$\mathcal{R}_{h,p}(u_{h,p}, v_{h,p}) = \mathcal{R}_{h,p}^{inv}(u_{h,p}, v_{h,p}) + \mathcal{R}_{h,p}^{visc}(u_{h,p}, v_{h,p}) + \mathcal{R}_{h,p}^{source}(u_{h,p}, v_{h,p}). \quad (2.9)$$

$\mathcal{R}_{h,p}^{inv}(u_{h,p}, v_{h,p})$, $\mathcal{R}_{h,p}^{visc}(u_{h,p}, v_{h,p})$ and $\mathcal{R}_{h,p}^{source}(u_{h,p}, v_{h,p})$ represent the contributions of the advective, diffusive and source terms to the weighted residual, respectively.
2.1.1 Inviscid discretization

The DG discretization of the inviscid flux term is given by:

\[
\mathcal{R}_{b,p}^{\text{inv}}(u,v) = - \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla v^T \cdot \mathcal{F}^{\text{inv}}(u) \, d\Omega \\
+ \sum_{e \in \Gamma_I} \int_e (v^+ - v^-)^T \mathcal{H}(u^+, u^-; n^+) \, dS \\
+ \sum_{e \in \Gamma_B} \int_e v^{+T} \mathcal{H}^B(u^+, u^B(u^+, BC); n^+) \, dS,
\]

where \((\cdot)^+\) and \((\cdot)^-\) denote the trace values evaluated from opposite sides of a face \(e\) and \(n^+\) is the unit normal vector pointing from the (+) side to (−) of a face. \(\Gamma_I\) and \(\Gamma_B\) represent the set of interior and boundary faces, respectively. \(\mathcal{H}\) and \(\mathcal{H}^B\) are the numerical flux functions on the interior and boundary faces respectively. For space-time problems, \(\mathcal{H}\) takes the form given in Eq. (2.11), where the contribution from the spatial components of the flux is given by the “spatial” numerical flux function \(\mathcal{H}_s\). Typically, \(\mathcal{H}_s\) uses an approximate Riemann solver such as Roe’s solver [51] for the Euler or Navier-Stokes equations. \(\mathcal{H}_s\) can also use the exact flux (i.e. Godunov’s flux [36]), as done for the Buckley-Leverett problem in Chapter 5. For space-time problems, the inviscid flux along the temporal dimension is always pointed in the direction of increasing time (i.e. to the future), in accordance with the laws of causality. Therefore it suffices for \(\mathcal{H}\) to perform a simple “temporal” upwinding where the temporal component of the inviscid flux is evaluated from the state at the earlier time.

\[
\mathcal{H}(u^+, u^-, n^+) = \begin{cases} 
\mathcal{H}_s(u^+, u^-, n^+) + \mathcal{F}_{d+1}^{\text{inv}}(u^+) \cdot n^+_t, & \text{if } n^+_t \geq 0 \\
\mathcal{H}_s(u^+, u^-, n^+) + \mathcal{F}_{d+1}^{\text{inv}}(u^-) \cdot n^+_t, & \text{otherwise}
\end{cases}
\]

where \(\mathcal{H}_s\) is the “spatial” numerical flux function and the space-time normal is decomposed into spatial and temporal components, \(n^+ = [n^+_s, n^+_t]\). \(\mathcal{F}_{d+1}^{\text{inv}}(u)\) represents the component of inviscid flux in the temporal direction. At the domain boundaries, the numerical flux \(\mathcal{H}^B\) is evaluated using a boundary state \(u^B\), which itself is a function
of both the interior state \( \mathbf{u}^+ \) and the user-specified boundary condition \( BC \).

### 2.1.2 Viscous discretization

In this thesis, the viscous flux terms are discretized using the second method proposed by Bassi and Rebay (BR2) [8, 9]. For simplicity of notation, the jump \([\cdot]\) and average \(\{\cdot\}\) operators are defined for a scalar \( s \) and a vector \( \mathbf{v} \) on an interior face as:

\[
\{s\} = \frac{1}{2}(s^+ + s^-), \quad \{\mathbf{v}\} = \frac{1}{2}(\mathbf{v}^+ + \mathbf{v}^-) \quad (2.12)
\]

\[
[s] = s^+ \mathbf{n}^+ + s^- \mathbf{n}^-, \quad [\mathbf{v}] = \mathbf{v}^+ \cdot \mathbf{n}^+ + \mathbf{v}^- \cdot \mathbf{n}^-
\]

The viscous discretization can then be written as follows:

\[
\mathcal{R}^{\text{visc}}_{h,p}(\mathbf{u}, \mathbf{v}) = \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \nabla \mathbf{v}^T \cdot (\mathbf{A}(\mathbf{u}) \nabla \mathbf{u}) \, d\Omega
\]

\[
- \sum_{e \in \Gamma_f} \int_{e} \left[ \mathbf{A}^T(\mathbf{u}) \nabla \mathbf{v} \right]^T \cdot \left[ \mathbf{u} \right] + \left[ \mathbf{v} \right]^T \cdot \left\{ \mathbf{A}(\mathbf{u}) (\nabla \mathbf{u} + \eta_f \mathbf{r}_f([\mathbf{u}])) \right\} \, dS
\]

\[
- \sum_{e \in \Gamma_B} \int_{e} \left[ (\mathbf{A}_B^T \nabla \mathbf{v}^+)^T \cdot (\mathbf{u}^+ - \mathbf{u}^B) \cdot \mathbf{n}^+ + (\mathbf{v}^+ \cdot \mathbf{n}^+)^T \cdot \mathbf{A}_B (\nabla \mathbf{u}^B + \eta_f \mathbf{r}_f((\mathbf{u}^+ - \mathbf{u}^B) \cdot \mathbf{n}^+)) \right] \, dS,
\]

where the boundary fluxes are set using \( \mathbf{u}^B(\mathbf{u}^+; BC) \), \( \mathbf{A}_B(\mathbf{u}^B; BC) \), and \( \nabla \mathbf{u}^B(\nabla \mathbf{u}^+; BC) \).

The lifting operator \( \mathbf{r}_f : [V_{h,p}(f)]^{d+1} \to [V_{h,p}]^{d+1} \), essentially penalizes jumps in the solution across a face, and is defined as follows for an interior face \( f \):

\[
\sum_{\kappa \in \kappa_f} \int_{\kappa} \tau^T \cdot \mathbf{r}_f(\mathbf{q}) \, d\Omega = - \int_{f} \{\tau\}^T \cdot \mathbf{q} \, dS, \quad \forall \tau, \mathbf{q} \in [V_{h,p}]^{d+1} \quad (2.14)
\]

where \( \kappa_f \) is the set of elements sharing the face \( f \). For boundary faces, the lifting operator is defined as:

\[
\int_{\kappa_B} \tau^T \cdot \mathbf{r}_f(\mathbf{q}) \, d\Omega = - \int_{f} (\tau^+ \cdot \mathbf{n}^+)^T \cdot \mathbf{q} \, dS, \quad \forall \tau, \mathbf{q} \in [V_{h,p}]^{d+1} \quad (2.15)
\]
where $\kappa_B$ is the element containing the boundary face. The stability of the DG discretization requires that the BR2 stabilization parameter, $\eta_f$, is greater than or equal to the number of faces in an element [28]. Hence in this work where triangular meshes are used, this parameter is set to $\eta_f = 3$.

### 2.1.3 Source discretization

The discretization of the source terms follows the formulation proposed by Bassi et al. in [7] where the state gradients are augmented with a global lifting operator as shown below.

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

(2.16)

where the global lifting operator $\mathbf{r}_g : V_{h,p} \to [V_{hp}]^d$ is defined as:

$$
\mathbf{r}_{\text{glob}}(u) = \sum_{f \in \Gamma_I} \mathbf{r}_f([u]) + \sum_{f \in \Gamma_B} \mathbf{r}_f((u^+ - u^B) \cdot n^+)
$$

(2.17)

This approach was also shown to be asymptotically dual-consistent by Oliver in [43]. 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 [42].

For the problems considered in this thesis, the source term only depends on the space-time coordinate $\hat{x}$ and the solution $u$. Since there is no dependence on the state gradient $\nabla u$, the global lifting operator is not required, but it has been added into the formulation for sake of completeness.

### 2.2 Solution method

In this work, the $V_{h,p}$ space defined in Eq. (2.7) is spanned by an element-wise discontinuous, nodal Lagrange polynomial basis. By expressing the solution $u_h$ and the test functions $v_h$ in terms of the selected basis, the discrete solution $U$ can be
obtained by solving the discrete system of equations,

\[ \mathbf{R}(\mathbf{U}) = 0 \]  \hspace{1cm} (2.18)

using Newton’s method, where \( \mathbf{R}(\mathbf{U}) \) is the discrete residual vector. The \( i \)-th entry of \( \mathbf{R}(\mathbf{U}) \) is the residual evaluated against the \( i \)-th basis function, \( \phi_i \), i.e. \( [\mathbf{R}(\mathbf{U})]_i = \mathcal{R}_{h,p}(U_j \phi_j, \phi_i) \). Given a discrete solution \( \mathbf{U}^k \), the solution update \( \Delta \mathbf{U}^k \) for the next Newton iteration is given by:

\[ \Delta \mathbf{U}^k = -\left( \frac{\partial \mathbf{R}}{\partial \mathbf{U}} \right|_{\mathbf{U}^k}^{-1} \mathbf{R}(\mathbf{U}^k) \]  \hspace{1cm} (2.19)

The initial solution vector \( \mathbf{U}^0 \) is obtained by performing an \( L^2 \)-projection of the initial space-time solution on to the \( V_{h,p} \) space. Since the problems considered in this work are relatively small, a sparse direct solver (UMFPACK [19]) was used to solve resulting linear system in Eq. (2.19).

### 2.2.1 Line search

In order to improve the robustness of the nonlinear solver, a line search algorithm is used to limit the solution update at each iteration. This is done by scaling \( \Delta \mathbf{U}^k \) by an update-fraction parameter, \( \eta \leq 1 \), as shown below:

\[ \mathbf{U}^{k+1} = \mathbf{U}^k + \eta \cdot \Delta \mathbf{U}^k \]  \hspace{1cm} (2.20)

The value of \( \eta \) is selected to ensure that the \( L^2 \)-norm of the residual vector strictly decreases from one iteration to the next. In the case of a system of conservation laws, \( \eta \) is chosen to ensure the reduction of all the sub-residual norms corresponding to each individual conservation law as well. The line-search algorithm for a scalar conservation law is summarized in Algorithm 1.
Algorithm 1 Halving line search algorithm
\[
\eta \leftarrow 1;
\]
\[
\tilde{U} \leftarrow U^k + \eta \cdot \Delta U^k;
\]
\[
\textbf{while } \left( \| R(\tilde{U}) \|_2 > \| R(U^k) \|_2 \text{ and } \eta \geq \eta_{\min} \right) \textbf{ do }
\]
\[
\eta \leftarrow \frac{\eta}{2};
\]
\[
\tilde{U} \leftarrow U^k + \eta \cdot \Delta U^k;
\]
\[
\textbf{end while}
\]
\[
\textbf{if } \eta \geq \eta_{\min} \textbf{ then}
\]
\[
U^{k+1} \leftarrow U^k + \eta \cdot \Delta U^k;
\]
\[
\textbf{else}
\]
\[
\text{return error;}
\]
\[
\textbf{end if}
\]

2.2.2 Continuation methods

It was observed that Newton’s method with a line search alone was not sufficient to solve some of the highly nonlinear problems considered in this thesis, especially for high-order discretizations. In such situations, continuation methods can be used to assist the nonlinear solver by sacrificing speed for increased robustness. This work employs two kinds of continuation methods: P-sequencing and pseudo-time continuation, and the instances in which they are used will be mentioned in later chapters.

P-sequencing

The main idea behind this continuation method is to use a lower-order solution as the initial guess for solving a higher-order problem. For example, if Newton’s method is struggling to directly converge to a P1 solution from the initial condition, an alternative approach may be to first obtain a P0 solution on the given mesh, project the P0 solution onto the P1 polynomial space, and then restart Newton’s method from the projected solution to obtain the P1 solution. Any high-order problem can be broken down in this manner into sub-problems of consecutively increasing polynomial order, where the solution of one sub-problem is projected and used as the starting point for
the next.

**Pseudo-time continuation**

This continuation method is commonly used to obtain steady-state solutions using an unsteady formulation. Given a discrete solution $U^n \in \mathbb{R}^N$, the solution one pseudo-timestep later, $U^{n+1}$, is obtained by solving the following system of equations:

$$
R_\tau \equiv M_\tau (U^{n+1} - U^n) + R(U^{n+1}) = 0,
$$

(2.21)

where $R_\tau(U)$ is the discrete pseudo-unsteady residual vector, and $M_\tau$ is the mass matrix weighted by the inverse of local elemental pseudo-timesteps, $\Delta \tau_\kappa$. At each pseudo-timestep, an iteration of Newton’s method is used to approximately solve Eq. (2.21), and the state update $\Delta U$ is computed as:

$$
U^{n+1} - U^n \approx \Delta U \equiv -\left(M_\tau + \frac{\partial R}{\partial U} \bigg|_{U^n}\right)^{-1} R(U^n)
$$

(2.22)

The solution is then advanced in pseudo-time using the computed state updates and a line search, until the $L^2$-norm of the residual vector, $\|R(U^n)\|_2$, is below a user-specified tolerance. In the context of space-time problems, where time is treated as an additional spatial dimension, it is difficult to find a physical interpretation of the pseudo-time $\tau$ used in this method, and hence the calculation of an appropriate local pseudo-timestep size is far from trivial. However for the purpose of this thesis, it has been sufficient to set $\Delta \tau_\kappa$ for each element as follows:

$$
\Delta \tau_\kappa = \theta h_\kappa, \quad \forall \kappa \in \mathcal{T}_h,
$$

(2.23)

where $\theta$ is a global CFL-like scaling parameter and $h_\kappa$ is a measure of the element’s size as given by:

$$
h_\kappa = \min_{f \in \partial \kappa} \left( \frac{V_\kappa}{A_f} \right),
$$

(2.24)
where $A_f$ is the area of face $f$ and the $V_\kappa$ is the volume of element $\kappa$. In this work, $\theta$ is set to $10^4$ at the start of each nonlinear solve, and then progressively increased or decreased based on the value of the update fraction required by the line search. If the residual norm can be reduced with $\eta = 1$, then $\theta$ is increased by a factor of 2 to speed up convergence. Conversely, if the line-search requires $\eta$ to be very small, i.e. $\eta < \eta_{\text{min}}$, then the problem is relaxed further by repeating the iteration with $\theta$ decreased by a factor of 10.

### 2.3 Output error estimation

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

$$J = J(u),$$

(2.25)

where $J : V \to \mathbb{R}$ is the output functional of interest and $u \in V$ 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 by using the discrete DG solution $u_{h,p} \in V_{h,p}$ as:

$$J_{h,p} = J_{h,p}(u_{h,p}),$$

(2.26)

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

$$\varepsilon_{\text{true}} = J - J_{h,p} = J(u) - J_{h,p}(u_{h,p}).$$

(2.27)

Since $\varepsilon_{\text{true}}$ cannot be directly computed, the goal of the output error estimation is to approximate this true error in the output functional. In this work, this is done using the dual-weighted residual (DWR) method proposed by Becker and Rannacher [10, 11].
2.3.1 Dual-weighted residual method

In DWR, the true output error can be represented as

\[ \varepsilon_{true} = J(u) - J_{h,p}(u_{h,p}) = -R_{h,p}(u_{h,p}, \psi), \]  \hspace{1cm} (2.28)

where \( \psi \in W \equiv V + V_{h,p} \) is the true adjoint solution that satisfies

\[ \overline{R}'_{h,p}(u,u_{h,p})(w, \psi) = \overline{J}'_{h,p}(u,u_{h,p})(w), \quad \forall w \in W. \]  \hspace{1cm} (2.29)

Here, \( \overline{R}'_{h,p}(u,u_{h,p}) : W \times W \to \mathbb{R} \) and \( \overline{J}'_{h,p}(u,u_{h,p}) : W \to \mathbb{R} \) are the mean-value linearizations defined as

\[ \overline{R}'_{h,p}(u,u_{h,p})(w,v) \equiv \int_0^1 R'_{h,p}[(1 - \theta)u + \theta u_{h,p}](w,v)d\theta \]  \hspace{1cm} (2.30)

\[ \overline{J}'_{h,p}(u,u_{h,p})(w) \equiv \int_0^1 J'_{h,p}[(1 - \theta)u + \theta u_{h,p}](w)d\theta \]  \hspace{1cm} (2.31)

where \( R'_{h,p}[z](\cdot, \cdot) \) and \( J'_{h,p}[z](\cdot) \) denote the Fréchet derivative of \( R_{h,p}(\cdot, \cdot) \) and \( J_{h,p}(\cdot) \) with respect to the first argument, and evaluated about \( z \).

However, the true adjoint \( \psi \) is not computable since it lives in an infinite dimensional space \( W \), and its computation requires the true primal solution \( u \) according to Eq. (2.29). Therefore in this work, the true adjoint solution is approximated by a finite dimensional adjoint \( \psi_{h,\hat{p}} \in V_{h,\hat{p}} \) that can be obtained by solving a dual problem that is linearized about \( u_{h,p} \):

\[ R'_{h,\hat{p}}(u_{h,p})(v_{h,\hat{p}}, \psi_{h,\hat{p}}) = J'_{h,\hat{p}}(u_{h,p})(v_{h,\hat{p}}), \quad \forall v_{h,\hat{p}} \in V_{h,\hat{p}} \]  \hspace{1cm} (2.32)

The DWR error estimate of the output is obtained by substituting this approximate adjoint into Eq. (2.28):

\[ \varepsilon_{true} \approx -R_{h,p}(u_{h,p}, \psi_{h,\hat{p}}) \]  \hspace{1cm} (2.33)
Note that the approximate adjoint $\psi_{h,\hat{p}}$ needs to exist in a space that is richer than that of the approximate primal solution $u_{h,p}$ (i.e. $V_{h,\hat{p}} \supset V_{h,p}$), or else the DWR estimate yields zero due to Galerkin orthogonality. For the work in this thesis, the polynomial order of the adjoint approximation was chosen to be one order higher than that of the primal solution, i.e. $\hat{p} = p + 1$.

In [55], Yano remarks that although the quality of the error estimates can be adversely affected by the dual problem linearization errors and the adjoint approximation errors, the resulting error estimates are still sufficiently accurate for mesh adaptation purposes.

### 2.3.2 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 $\eta_\kappa$, associated with element $\kappa$, is obtained by element-wise restriction of the adjoint weight as follows:

$$\eta_\kappa \equiv |R_{h,p}(u_{h,p}, \psi_{h,\hat{p}}|_{\kappa})|$$  \hspace{1cm} (2.34)

By arguments of Galerkin orthogonality, Yano [55] shows that this localized error estimate represents a weighted product of the local primal error and the local adjoint error, which implies that the mesh adaptation process needs to account for the behaviors of both the primal and adjoint solutions.

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

$$\varepsilon \equiv \sum_{\kappa \in T_h} \eta_\kappa$$  \hspace{1cm} (2.35)
2.4 Mesh adaptation

After localized output error estimates have been obtained using the DWR method, the goal of mesh adaptation is to use this 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 [33, 1, 17]. But reservoir flows often contain features such as contact discontinuities (saturation shocks), which can be captured much more efficiently using 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, rather than just the measure of their size $h$.

2.4.1 Continuous mesh framework

The anisotropic information of an element $\kappa$ can be represented using a metric tensor $\mathcal{M}_\kappa$, which is a symmetric positive definite (SPD) matrix [52]. This metric tensor can be interpreted as a straightforward 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 $\{\mathcal{M}(x)\}_{x \in \Omega}$ 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 $\{\mathcal{M}(x)\}_{x \in \Omega}$. The length of a segment $\overrightarrow{ab}$ from point $a \in \Omega$ to point $b \in \Omega$ under the metric is given by:

$$l_{\mathcal{M}}(\overrightarrow{ab}) = \int_0^1 \sqrt{\overrightarrow{ab}^T \mathcal{M}(a + \overrightarrow{abs}) \overrightarrow{ab}} \, ds.$$  \hfill (2.36)

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 [55]. 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.

\[ \text{Mesh Generation} \]

\[ \text{Metric Field } \{ \mathcal{M}(x) \}_{x \in \Omega} \]

\[ \text{Implied Metric} \]

\[ \text{Mesh } \mathcal{T}_h \]

Figure 2-1: Mesh metric-field duality (Modisette [39])

### 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 [55], 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 this problem involves finding the optimal triangulation \( \mathcal{T}_h^* \) given by:

\[
\mathcal{T}_h^* = \arg \inf_{\mathcal{T}_h} \mathcal{E}(\mathcal{T}_h) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{T}_h) \leq \text{dof}_{\text{target}},
\]

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 \( \mathcal{T}_h \). Since the triangulation \( \mathcal{T}_h \) is defined by node coordinates and node connectivity, the optimiza-
tion problem presented above is a discrete-continuous optimization problem, that 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 [37], where a continuous Riemannian metric field, \( \mathcal{M} \equiv \{ \mathcal{M}(x) \}_{x \in \Omega} \) is optimized instead of the discrete mesh. This relaxed optimization problem involves finding an optimal metric field, \( \mathcal{M}^* \), where:

\[
\mathcal{M}^* = \arg \inf_{\mathcal{M}} \mathcal{E}(\mathcal{M}) \quad \text{s.t.} \quad \mathcal{C}(\mathcal{M}) \leq \text{dof}_{\text{target}} \quad (2.38)
\]

In order to apply a DOF constraint, the cost functional \( \mathcal{C}(\mathcal{M}) \) needs to be defined as:

\[
\mathcal{C}(\mathcal{M}) = \int_{\Omega} c_p \sqrt{\det(\mathcal{M}(x))} dx, \quad (2.39)
\]

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 \( \eta_\kappa \), and that each local contribution \( \eta_\kappa \) is also a function of the elemental metric tensor \( \mathcal{M}_\kappa \). The error functional \( \mathcal{E} \) can then be approximated as:

\[
\mathcal{E}(\mathcal{M}) \approx \sum_{\kappa \in \mathcal{I}_h} \eta_\kappa(\mathcal{M}_\kappa) \quad (2.40)
\]

To complete the problem statement, a definition of the local error function \( \eta_\kappa(\mathcal{M}_\kappa) \) is needed, but since a general analytical expression is unknown, a surrogate model of this function is 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 element $\kappa \in \Omega$, 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 $\kappa$. The original configuration of element $\kappa$ is denoted by $\kappa_0$. Figure 2-2 shows an example of the split configurations used for a 2D element, and the implied metric tensors $\mathcal{M}_{\kappa_i}$ associated with each configuration.

Figure 2-2: Example split configurations with associated metric tensors (Yano [55])

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

$$
\mathcal{R}_{h,p}^{\kappa_i}(\mathbf{u}_{h,p}^{\kappa_i}, \mathbf{v}_{h,p}^{\kappa_i}) = 0, \quad \forall \mathbf{v}_{h,p}^{\kappa_i} \in V_{h,p}(\kappa_i) \tag{2.41}
$$

where the local semi-linear form $\mathcal{R}_{h,p}(\cdot, \cdot)$ imposes boundary fluxes on $\kappa_i$ by assuming that the solution on neighboring elements does not change. Next, a localized DWR error estimate associated with the configuration $\kappa_i$ is computed as:

$$
\eta_{\kappa_i} \equiv |\mathcal{R}_{h,p}(\mathbf{u}_{h,p}^{\kappa_i}, \psi_{h,\hat{p}}|_{\kappa})|	ag{2.42}
$$

where $\hat{p} = p + 1$ as before.
Finally, the local metric $\mathcal{M}_{\kappa_i}$ associated with configuration $\kappa_i$ is obtained by computing an affine-invariant average of the implied metric tensors of each sub-element in $\kappa_i$ [45]. 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 $\eta_{\kappa}(\cdot) : \text{Sym}_d^+ \rightarrow \mathbb{R}^+$ aims to capture how the local error is affected by local changes to the metric field. For this purpose, a step tensor, $S_{\kappa_i}$, is defined to characterize the change in the metric tensor from configuration $\kappa_0$ to $\kappa_i$. This measurement is based on Pennec’s affine-invariant framework [45] as follows:

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

(2.43)

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:

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

(2.44)

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

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

(2.45)

where $R_{\kappa} \in \text{Sym}_d$ is a “rate” tensor that is synthesized from the known data by performing a least-squares regression as follows:

$$R_{\kappa} = \arg \min_{Q \in \text{Sym}_d} \sum_{i=1}^{n_{\text{config}}} (f_{\kappa_i} - \text{tr}(Q S_{\kappa_i}))^2$$

(2.46)
Finally, the local error model can be written in terms of $S_\kappa$ as:

$$\eta_\kappa(S_\kappa) = \eta_\kappa \exp(\text{tr}(R_\kappa S_\kappa))$$

(2.47)

**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. $S_v \in \text{Sym}_d$ 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. (2.38) can be solved using a gradient-based optimization algorithm as shown by Kudo [34].

The resulting optimal vertex-based metric can then passed into a metric based mesh generator to generate a new mesh that conforms to the optimized metric field. All adapted meshes used in this thesis were generated using the Bidimensional Anisotropic Mesh Generator (BAMG) developed by INRIA [29].
Chapter 3

Basic Concepts and Definitions

This chapter reviews some of the basic concepts involved with flows in porous media and discusses the key parameters that will be used to define the problems investigated in later chapters.

3.1 Representative elementary volume

A porous medium is a material that contains voids or “pores”. The skeletal portion of the material is referred to as the “matrix” and is usually a solid (e.g. rock, sand, clay). The pores can be of different sizes and are fully or partially filled with fluids (e.g. air, water, oil). These pores form the pathways for the transport of fluid through the porous medium.

The microscopic size of the pores makes it impossible to completely describe the pore structure of the material. This can be easily understood by noting the difference in scale between the subsurface of a typical reservoir, which may be in the scale of kilometres, and the individual pores in the underlying rock, which may only be a few micrometres across. Therefore, a continuum approach is used to model the porous medium where the important microscopic properties of the porous medium are averaged over a representative elementary volume (REV). This averaging process smears out the microscopic properties of the porous medium and replaces them with a new set of parameters, which are only present in the macroscale. Parameters such
as porosity, permeability and saturation are examples of such quantities, and will be discussed further below.

The choice of the REV’s size is arbitrary and problem dependent. In most cases, it is chosen to be large enough to average out the effects of individual pores, but also small enough to capture inhomogeneities in the porous medium, such as the presence of fractures. This work assumes that the macroscopic properties have been obtained using a REV of appropriate size.

3.2 Porosity

The porosity $\phi$ quantifies the ratio of empty space to solid material in a porous medium. In particular, it is defined as:

$$\phi = \frac{\text{volume of pore space in the REV}}{\text{volume of the REV}} \quad (3.1)$$

Note that $\phi$ is a dimensionless quantity that takes values between 0 and 1. The pore space can also be categorized into “connected pores” and “dead-end pores”, where the latter represents pores that cannot contribute to the flow of fluid. Hence it is useful to define an effective porosity $\phi_{\text{eff}}$ as:

$$\phi_{\text{eff}} = \frac{\text{volume of connected pores in the REV}}{\text{volume of the REV}} \quad (3.2)$$

Since the effective porosity is a more important parameter for characterizing flows in porous media, all subsequent references to porosity or $\phi$ in this thesis actually refer to the effective porosity.

3.3 Saturation

When multiple fluids or phases flow through a porous medium, each phase occupies a certain fraction of the pore space. The volume fraction occupied by a given phase is known as its saturation and is an important quantity of interest in reservoir flows.
The saturation of phase $\alpha$ is given by:

$$S_\alpha = \frac{\text{volume of fluid } \alpha \text{ in the REV}}{\text{volume of connected pore space in the REV}}$$  \hspace{1cm} (3.3)

Furthermore, the assumption that all fluid phases completely fill up the pore space yields the following closure relation:

$$\sum_{\alpha} S_\alpha = 1$$  \hspace{1cm} (3.4)

### 3.4 Darcy’s law and permeability

One of the most used laws in this work, and reservoir flows in general, is Darcy’s law formulated by Henry Darcy based on the results of his experiments in 1856 [18]. Although initially used to describe the unidirectional flow of a single fluid through a porous medium, it has since been generalized to multi-dimensional and multi-phase flows. Darcy’s law for a one-dimensional flow is given by:

$$Q = -\frac{K A \Delta p}{\mu L},$$  \hspace{1cm} (3.5)

where $Q$ is the volumetric flow rate of the fluid, $K$ is the permeability of the porous medium, $\mu$ is the viscosity of the fluid, $A$ is the cross-sectional area of the porous medium through which the fluid is flowing, $L$ is the length of the medium measured parallel to direction of flow, and $\Delta p$ is the pressure differential measured across length $L$. The differential form of the Eq. (3.5) can be written as:

$$u = -\frac{K}{\mu} \frac{\partial p}{\partial x},$$  \hspace{1cm} (3.6)

where $u$ is the Darcy velocity.

The (absolute) permeability $K$ quantifies the fluid conductivity of the porous medium, where a large permeability value implies that the medium readily transmits fluids. Since $K$ is solely a property of the porous medium, it is also referred to as the
intrinsic permeability. In multi-dimensional flow, permeability is often represented as a tensor $K$ since the conductivity of the porous medium can vary with the direction of flow. If the medium is isotropic, the permeability tensor reduces to a scaled identity tensor. Permeability is typically given in units of darcies (or milli-darcies), where 1 darcy = $9.869 \times 10^{-13}$ m$^2$. Since permeability is a property of the porous medium, it can be modeled as a function of the spatial coordinate, $K = K(x)$. In the case of heterogeneous reservoirs, the permeability value may vary sharply in space and span multiple orders of magnitude due to the different types of rock present. However, for the scope of this work, it is assumed that the permeability field is constant in space (i.e. homogeneous reservoirs).

Eq. (3.6) can also be generalized to account for multiple dimensions, multiple phases and elevation changes as follows:

$$u_\alpha = -\frac{K k_{r\alpha}}{\mu_\alpha} (\nabla p_\alpha - \rho_\alpha g),$$  \hspace{1cm}  (3.7)

where $u_\alpha$ is the Darcy velocity vector for phase $\alpha$, $K$ is the permeability tensor, $k_{r\alpha}$ is the relative permeability, $p_\alpha$ is fluid pressure, $\rho_\alpha$ is the fluid density, and $g$ is the gravity vector. The subscript $\alpha$ implies that the quantity is related to phase $\alpha$.

In the context of multi-phase flows, each phase $\alpha$ that is being transported through the porous medium has a relative permeability $k_{r\alpha}$, which measures how the presence of other phases hinders the flow of phase $\alpha$. For example, if $S_\alpha$ is small it means that other phases are filling up most of the available pore space, and therefore it is hard for phase $\alpha$ to move. This implies that $k_{r\alpha}$ must also be small. The interfacial tension between phases may also further hinder the flow of each phase present in the flow.

Although the exact form of the relative permeability functions cannot be found analytically, there exist some simplified empirical relations such as the “power law” relations [35] given below for a wetting phase $w$ and a non-wetting phase $n$:

$$k_{rw}(S_w) = S_w^\lambda$$ \hspace{1cm} (3.8)
\[ k_{rn}(S_w) = (1 - S_w)^\lambda \]  
(3.9)

where \( \lambda \) is typically assumed to be some even power. Brooks and Corey also introduce a similar but slightly more sophisticated model in [12]:

\[ k_{rw}(S_w) = \frac{S_w^{2+\lambda}}{\lambda} \]  
(3.10)

\[ k_{rn}(S_w) = (1 - S_w)^2 \left(1 - \frac{S_w^{2+\lambda}}{\lambda}\right) \]  
(3.11)

This work assumes the form given in Eq. (3.8) and (3.9), with \( \lambda = 2 \) or 4.

### 3.5 Capillary pressure

At the microscopic scale, the interface between two phases is generally curved. It can be then shown using arguments of mechanical equilibrium that there exists a difference between the non-wetting phase pressure \( (p_n) \) and the wetting phase pressure \( (p_w) \). This pressure difference is referred to as the capillary pressure, \( p_c \),

\[ p_c = p_n - p_w \]  
(3.12)

and it depends on the interfacial (surface) tension and the geometry of the pore space.

However, since information about individual pore sizes are not available at the macro-scale (due to averaging), the capillary pressure needs to be described and modeled in terms of a macroscopic quantity. Most often, capillary pressure is modeled as a function of the wetting phase saturation, and the experimentally determined function \( p_c(S_w) \) can be thought to incorporate information about the pore sizes. One of the simplest models for capillary pressure is the linear relationship given below, and will be used in this work.

\[ p_c(S_w) = p_{cmax} (1 - S_w) \]  
(3.13)

\( p_{cmax} \) is the maximum capillary pressure, which is usually determined experimentally.
A more complex model due to Brooks and Corey [12] is given by:

\[ p_c(S_w) = p_e S_w^{-\frac{1}{\lambda}} \]  

(3.14)

where \( p_e \) represents the *entry pressure*, which is the minimum pressure required for the non-wetting phase to start displacing the wetting phase. The \( \lambda \) parameter represents the pore size distribution of the porous medium, where large values of \( \lambda \) describe a uniform medium with similar pore sizes, and small values of \( \lambda \) describe a non-uniform material with a wide range of pore sizes.

### 3.6 Compressibility

The compressibility \( c \) of a fluid \( \alpha \) under isothermal conditions is defined as:

\[ c_\alpha = \frac{1}{\rho_\alpha} \frac{\partial \rho_\alpha}{\partial p} \bigg|_T \]  

(3.15)

For reservoir flow problems, it can be often assumed that the fluid compressibility and temperature are constant over the pressure ranges of interest [4, 44], and hence Eq. (3.15) can be integrated to produce:

\[ \rho_\alpha(p) = \rho_{\alpha_{\text{ref}}} e^{c_\alpha(p - p_{\text{ref}})} \]  

(3.16)

where \( \rho_{\alpha_{\text{ref}}} \) is the density at the reference pressure \( p_{\text{ref}} \). Although somewhat counter intuitive, it is certainly possible for the pore volume itself to expand or shrink under pressure, and therefore this leads to the definition of a rock compressibility \( c_\phi \):

\[ c_\phi = \frac{1}{\phi} \frac{\partial \phi}{\partial p} \bigg|_T \]  

(3.17)

where \( \phi \) is the rock porosity. By assuming \( c_\phi \) and temperature are constant, this expression can be integrated as before to produce:

\[ \phi(p) = \phi_{\text{ref}} e^{c_\phi(p - p_{\text{ref}})} \]  

(3.18)
All compressible flow problems in this thesis assume constant compressibilities and isothermal conditions, and therefore obey Eq. (3.16) and (3.18).
Chapter 4

Single-Phase Flow

This chapter introduces the simplest type of reservoir flow, which is the flow of a single fluid through a porous medium, and then proceeds to apply the previously described solution adaptive framework to a simple test problem. Section 4.2 describes how the governing equations of the single-phase flow problem can be recast into the space-time formulation that is used in this work. Finally, the space-time adaptive results from our approach are presented and analyzed in comparison to those of a conventional time-marching finite volume method.

4.1 Governing equations

The general form of Darcy’s law given in Eq. (3.7) simplifies for a single-phase flow as follows:

$$u = -\frac{K}{\mu} (\nabla p - \rho g),$$

(4.1)

where the phase subscripts are no longer needed and the relative permeability $k_r$ has trivially been set to 1 for a single-phase flow. Then by performing a mass balance across a control volume, a statement of mass conservation can be derived for the fluid
flowing through the porous medium:

\[
\frac{\partial (\rho \phi)}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \rho q,
\]  

(4.2)

where \( \phi \) is the porosity of the porous medium and \( q(x, t) \) represents a source/sink term which is specified as a volumetric flow rate per unit volume (positive \( q \) indicates mass injection).

For the work in this thesis, Eq. (4.2) is further simplified using the following assumptions:

- Negligible gravitational effects
- Slightly compressible fluid \( (c_f \ll 1) \)
- Isothermal reservoir conditions
- Constant porosity field \( (c_r = 0) \)
- Constant permeability field \( K \)
- Constant fluid viscosity \( \mu \)

By using the expression for the Darcy velocity given in Eq. (4.1) and applying the assumptions above, the mass conservation equation can be simplified to produce the single-phase flow equation (or “pressure” equation) [44]:

\[
\phi c_f \frac{\partial p}{\partial t} + \nabla \cdot \left( -\frac{K}{\mu} \nabla p \right) = q,
\]  

(4.3)

where \( p(x, t) \) is the fluid pressure. Under the assumptions given above, Eq. (4.3) is a linear parabolic equation.
4.2 Space-time formulation

The pressure equation given in Eq. (4.3) is in the form of a \( d \)-dimensional unsteady conservation law, and therefore can be solved using conventional time marching methods. However, in this work, we recast the \( d \)-dimensional unsteady conservation law as a \((d+1)\)-dimensional “steady-state” conservation law, where the temporal dimension is treated like an additional spatial dimension. In this perspective, the time-derivative term in Eq. (4.3) can be interpreted as the derivative of a convective flux in the \((d + 1)^{th}\) dimension. In particular, the space-time formulation of the pressure equation that is consistent with Eq. (2.2) is given as follows:

\[
\sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} F^\text{inv}_j(p) - \sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} F^\text{visc}_j(p, \nabla p) = q(\hat{x}),
\]

(4.4)

where,

\[
F^\text{inv}(p) = \left( 0, \phi c_f p \right) \quad \text{and} \quad F^\text{visc}(p, \nabla p) = \left( \left[ \frac{K}{\mu} \nabla x p \right]^T, 0 \right).
\]

(4.5)

The augmented “space-time” coordinate is given by \( \hat{x} = [x, t] \), and \( \nabla x \) signifies that the derivatives are taken only with respect to the spatial dimensions. As shown in Eq. (4.5), the first \( d \) columns of \( F^\text{inv} \) and \( F^\text{visc} \) represent the standard spatial fluxes, and the last column contains the flux in the temporal dimension.

Eq. (4.4) is now in the form that can be solved using the methods described in Chapter 2. Since the unsteady problem is solved for a finite time interval \( I = [0, T] \), the domain of the space-time problem contains two temporal boundaries. The initial condition of the original unsteady problem is transformed into a Dirichlet boundary condition along the \( t = 0 \) boundary of the space-time domain. The \( t = T \) boundary can be interpreted as an “outflow” boundary since information only propagates out of it, obeying the law of causality. Hence, at this boundary, it is sufficient to set the boundary state \( u^B \) equal to the trace value from the interior, \( u^+ \).
4.3 Test problem

This section presents a simple 1D single-phase flow problem, which is then formulated as a 2D space-time problem as described above, and solved using the adaptive framework given in Chapter 2.

4.3.1 Problem statement

As shown by the schematic in Figure 4-1, the problem involves extracting oil from a one-dimensional reservoir. The length of the reservoir is given by $L = 1000$ ft, and it is assumed to be initially completely filled with oil at a constant pressure of 2500 psi. An extraction well of length $L_s = 40$ ft situated at the right end of the reservoir extracts oil at a constant rate of $Q = 100$ barrels per day. The well is started at $t = 0$ and is shut off after 10 days. The reservoir is assumed to have a constant width $\Delta y$ and constant depth $\Delta z$, so that the oil extraction rate can be specified as a volumetric flow rate, as in a real reservoir.

![Schematic of 1D reservoir](image)

Figure 4-1: Schematic of 1D reservoir

The governing equation is the 1D pressure equation:

$$
\phi c_f \frac{\partial p}{\partial t} - \frac{K}{\mu} \frac{\partial^2 p}{\partial x^2} = -q_p(x,t), \quad \forall x \in [0, L], \ t \in [0, T], \quad (4.6)
$$

with the initial condition:

$$
p(x, 0) = 2500 \text{ psi}, \quad (4.7)
$$
and no flow boundary conditions at the left and right ends of the reservoir, which reduce to the following Neumann conditions:

\[
\frac{\partial p}{\partial x}\bigg|_{x=0} = \frac{\partial p}{\partial x}\bigg|_{x=L} = 0. \tag{4.8}
\]

Other relevant parameters are given below:

\[
\begin{align*}
\phi &= 0.2, & c_f &= 1.5 \times 10^{-5} \text{ psi}^{-1}, \\
K &= 200 \text{ md}, & \mu &= 1 \text{ cP}, \\
Q &= 100 \text{ bbls/day}, & L_s &= 40 \text{ ft}, \\
\Delta y &= 100 \text{ ft}, & \Delta z &= 30 \text{ ft}.
\end{align*}
\]

The extraction well is modeled by the source term \(q_p(x,t)\), which acquires a value of \(\frac{Q}{L_s \Delta y \Delta z}\) when the well is on, and zero otherwise. Jumps in the source term are smoothed out using a smoothstep function that spans 10 ft spatially and 0.3 days in time. Note that appropriate unit conversions were also applied to the quantities given above to make the pressure equation dimensionally consistent. Figure 4-2 shows the domain of the space-time problem that is solved using the adaptive framework.

![Figure 4-2: Schematic of space-time domain](image)
4.3.2 Objectives

The extraction of oil causes the pressure around the extraction well to decrease. It is important to ensure that the oil pressure does not decrease below the bubble point pressure of oil, because below this pressure, any gases that are dissolved in the oil start to evolve out and form bubbles. The formation of gas near an extraction well poses a variety of problems, which include safety risks and reduced oil production rates due to the excess gas that is produced. The ability to accurately predict the minimum pressure around an extraction well enables the well shut-off times to be planned with minimal risk. However in this simple test problem, the behavior of the well is pre-defined, and the goal is simply to find the most accurate estimate of the minimum oil pressure for a given amount of computational resources. The specific objectives of this test problem are listed below:

- Determine the oil pressure behavior in the reservoir up to $T = 30$ days.
- Accurately predict the magnitude of minimum pressure in the reservoir.
- Compare results with those from a finite volume method that is second order accurate in space, and first order accurate in time.

The output of interest as defined in Eq. (2.25) was chosen to be:

$$ J = \int_{\Omega} \frac{1}{p^r} \, d\Omega, $$

(4.9)

where $r > 0$, and $\Omega$ is the space-time domain. The adaptation process strives to find the best estimate for $J$, and due to the inverse relationship between the pressure $p(x, t)$ and $J$, it also means that the adaptive method will target regions that affect lower pressures. In this work, $r$ was set to 2.
4.4 Numerical results

4.4.1 Adapted solutions

Figure 4-3 shows a piecewise quadratic (P2) adapted space-time solution containing approximately 50,000 degrees of freedom (DOF). The pressure at the right end of the reservoir starts to decrease from the constant initial condition of 2500 psi as expected. Once the extraction well is shut off at $t = 10$ days, the pressure recovers and settles to a constant value of around 1900 psi. The region of minimum pressure that is of interest occurs just before the well is shut off, and is depicted by the blue region in the figure.

![Figure 4-3: Adapted space-time primal solution - P2 50k DOF](image)

The final optimal mesh produced by the adaptation algorithm to obtain the solution above is shown in Figure 4-4. The instance shown is after 15 iterations of the adaptive algorithm. The adaptation process focuses on the minimum pressure region, as seen by the large clustering of elements in that region. The dense horizontal
clustering of elements at $t = 10$ is produced to accurately capture the large temporal pressure gradients that occur when the well is shut-off. Similarly, large temporal gradients in the initial pressure transient drive the adaptation occurring along the bottom-right boundary of the space-time domain. The vertical clustering of elements seen at around $x = 960$ is due to the transition from an inactive source term on the left to an active source term on the right.

![Figure 4-4: Adapted space-time mesh - P2 50k DOF](image)

The corresponding adjoint solution of the single-phase problem is given in Figure 4-5. Although the apparent plain nature of the adjoint solution seems to suggest that the adaptation is mainly driven by features in the primal solution, a more careful study of the solutions show that this is not entirely the case. The influence of the adjoint solution is much more evident in piecewise linear (P1) adapted solutions, and will be discussed further in Section 4.4.3.
4.4.2 Comparisons with a conventional method

This section compares the results obtained with the space-time adaptive DG method to those from a conventional time-marching finite volume method. The FV method is second-order accurate in space and first-order accurate in time, and is representative of the solvers that are being widely used in industry.

Figure 4-6 shows the pressure histories at the right end of the reservoir, as predicted by the two methods up to 30 days. The pressure curves of the FV method and the adapted space-time DG method (P1 and P2) are consistent with each other, with the pressure decreasing up to $t = 10$ days and then abruptly recovering to a constant value. However, upon zooming in to the region of minimum pressure, as shown in Figure 4-7, the P1 DG curve shows a slight deviation from the P2 DG and FV curves.

Although the P2 DG and FV results appear virtually identical, the two solutions contain a significant difference in computational cost. The FV solution was obtained with a timestep of $\Delta t = 5 \times 10^{-5}$ days, which corresponds to about 600,000 timesteps.
Figure 4-6: Pressure history at $x = 1000$ ft

Figure 4-7: Zoomed-in view of pressure history at $x = 1000$ ft

over the 30 days. In contrast, the adapted DG solutions contain $\sim 50,000$ degrees of freedom in the space-time solution, which means that the final adapted P2 solution only has about 130 high-order elements along the right boundary of the space-time domain. The fact that a finely resolved low-order solution is well matched by a rather coarse high-order solution clearly displays the potential of high-order adaptive
methods. But since the metrics of performance and efficiency may differ according to the application and the users’ needs, a more careful study is required to compare the performance of the different numerical methods considered. The rest of this section investigates two of the commonly used metrics, error-to-DOF and error-to-runtime efficiency.

**Error-to-DOF efficiency**

Figure 4-8 shows how the output error, $\mathcal{E} \equiv J - J_{h,p}$, behaves with the number of spatial degrees of freedom in the solution for three different numerical methods: time-marching FV, space-time DG with uniformly refined meshes, and space-time DG with output-based adaptation.

![Figure 4-8: Error vs. $h$](image-url)
The true output $J$ was estimated by solving the problem on a very fine structured space-time mesh with P2 elements. The numbers next to each data point on the figure represent the number of space-time DOF in that solution, except for the FV results where the numbers show the number of spatial cells used in the simulation. Note that the horizontal axis scales with the average size of an element in the mesh, $h$, to ensure a fair comparison between the time-marching FV results and the space-time DG results. Eq. (4.10) shows how $h$ was estimated for the two methods:

$$
\frac{1}{h} = \begin{cases} 
\text{number of spatial cells,} & \text{for time-marching FV} \\
\sqrt{\text{(Total space-time DOF)}}, & \text{for space-time DG}
\end{cases}
$$  \hspace{1cm} (4.10)

The dotted lines in the figure represent the time-marching FV results, obtained with two different timestep sizes. Both curves seem to flatten out relatively quickly when the number of cells is increased. This is because the error incurred by the first order temporal discretization starts to dominate, and the total error does not decrease further even though the error contribution from the spatial discretization is reduced.

The green and brown solid lines represent space-time DG results obtained with uniformly refined structured meshes, for P1 and P2 respectively. These curves do not flatten out because the uniform refinement of the space-time mesh reduces both the spatial and temporal errors. Furthermore, these results benefit from the high-order accuracy in both space and time, which allow the space-time DG method to achieve errors that are more than two orders of magnitude smaller than the finest FV result. The uniformly refined space-time DG results also appear to obtain the expected super-convergence rates of $O(h^{2p})$, since the green P1 line has a slope of approximately 2, and the brown P2 line has a slope closer to 4.

The magenta and orange solid lines represent adapted space-time DG results, for P1 and P2 respectively. It is clear from the figure that for cases with a sufficiently large number of DOF, the adapted results achieve the same level of error as the uniformly refined results, but with fewer DOF as expected.
Error-to-runtime efficiency

This sub-section compares the runtime efficiency of the different numerical methods, which may be of greater practical importance than certain other comparisons, such as the DOF efficiency discussed above. Figure 4-9 shows the total time (in seconds) taken by each method to achieve an output error $\mathcal{E}$. Note that the runtime of an adaptive result includes the time taken to generate multiple intermediate meshes, solve on those meshes and also perform error estimation. All simulations were run on a single thread of an Intel Core i7-3630QM (2.4GHz) processor, with a cache size of 6MB and 16GB of RAM.

Unlike in the comparison of DOF efficiency, this metric clearly separates the three sets of results, with the uniformly refined space-time DG results being the most efficient in terms of runtime. Furthermore, the higher order P2 result (brown solid
line) appears to be the most efficient for lower error levels. The time-marching FV results in dashed lines are about an order of magnitude slower in achieving the same level of error, compared to the uniformly refined space-time results. This is mainly due to the large number of timesteps that is required to maintain a reasonable temporal accuracy.

The adapted space-time results are nearly an order of magnitude slower than even the FV results. This is a direct consequence of the adaptive algorithm having to solve on multiple intermediate adapted meshes before reaching the final solution. The adapted results shown in the figure are after 15 adaptation iterations, i.e. 15 new meshes were generated and solved. Unlike the space-time adaptive method, the time-marching FV and uniformly refined space-time DG methods do not compute output error estimates.

Finally, it can be argued that the relatively smooth nature of this problem, and the lack of interesting solution features means that the adaptive method does not have a significant advantage over uniform mesh refinement, in terms of reducing error. Taking into account the significant overhead of multiple mesh generation and error estimation, an adaptive method is not the most attractive option for solving this linear problem. It is expected that the adaptive method is better suited for problems with nonlinearities, heterogenous properties or other dominant solution features. However, the higher order accuracy in space and time is indeed a desirable property, as verified by the superior performance of the uniformly refined space-time DG results in both efficiency metrics.

4.4.3 Other observations

The nature of the optimal mesh for a given solution polynomial order may not always be intuitive, even for simple problems such as this single-phase flow problem. It is also possible for the optimal mesh for one $p$-order to be radically different from the optimal mesh for a different $p$-order, in a way that is impossible to predict beforehand. This section presents an example of a similar case where the P1 optimal mesh is significantly different from the higher order optimal meshes.
Figure 4-10 shows the final adapted mesh produced by the adaptive algorithm using piecewise linear (P1) space-time elements. This mesh contains a large number of diagonal striations towards the lower-left corner of the space-time domain, which are not observed in the P2 adapted mesh given in Figure 4-11.

Figure 4-10: Adapted space-time mesh - P1 50k DOF

Figure 4-11: Adapted space-time mesh - P2 50k DOF
As mentioned before, the need for these striations only in the P1 mesh is not entirely clear from the primal or adjoint solutions itself (Figures 4-3 and 4-5), which are smooth in the region concerned. The presence of the striations in the P1 solution but not in the P2 solution implies that the solution contains a feature that is easily captured with a few piecewise quadratic or higher order elements, but not with piecewise linear elements. This also probably means that the striations are driven by large second derivatives in the solution. Normalized plots of $\frac{\partial^2 p}{\partial x^2}, \frac{\partial^2 p}{\partial t^2}$ and $\frac{\partial^2 p}{\partial x \partial t}$, given in Figures 4-12 to 4-14, contain large values near the lower-right end of the space-time domain, masking any smaller variations that maybe present near the striations. Hence, zoomed-in plots of $\frac{\partial^2 p}{\partial x^2}, \frac{\partial^2 p}{\partial t^2}$ and $\frac{\partial^2 p}{\partial x \partial t}$ were also analyzed on the same scale, as shown in Figures 4-15 to 4-17, but they also do not contain any significant evidence for the cause of the striations.

Figure 4-12: Second spatial derivative of adapted primal solution - $\left| \frac{\partial^2 p}{\partial x^2} \right| \times L^2$
Figure 4-13: Second time derivative of adapted primal solution - $\left| \frac{\partial^2 p}{\partial t^2} \right| \times T^2$

Figure 4-14: Second cross derivative of adapted primal solution - $\left| \frac{\partial^2 p}{\partial x \partial t} \right| \times LT$
Figure 4-15: Zoomed-in view of $\left| \frac{\partial^2 \rho}{\partial x^2} \right| \times L^2$ with overlaid mesh

Figure 4-16: Zoomed-in view of $\left| \frac{\partial^2 \rho}{\partial t^2} \right| \times T^2$ with overlaid mesh
Similarly, the normalized second derivatives of the adjoint solution, $\frac{\partial^2 \psi}{\partial x^2}$, $\frac{\partial^2 \psi}{\partial t^2}$ and $\frac{\partial^2 \psi}{\partial x \partial t}$ (Figures 4-18 to 4-20), were also inspected to find a possible cause for the striations. The relatively large magnitude of $\frac{\partial^2 \psi}{\partial t^2}$ near the right boundary on Figure 4-19 obscures any variations that may be present around the region of interest. Therefore, zoomed-in plots of $\frac{\partial^2 \psi}{\partial x^2}$, $\frac{\partial^2 \psi}{\partial t^2}$ and $\frac{\partial^2 \psi}{\partial x \partial t}$ are given in Figures 4-21 to 4-23, on the same scale for easier comparisons. The second temporal derivative of the adjoint solution, $\frac{\partial^2 \psi}{\partial t^2}$, contains the largest variations that seem to match the presence of the striations in the P1 mesh, as shown in Figure 4-22. Note, in order to obtain higher order derivatives, the solution on the P1 adapted mesh was resolved using P4 elements.

It is therefore clear that the adaptation is influenced by features in the primal solution as well as the adjoint solution. These observations also further reinforce the advantages of automated mesh adaptation techniques, which are able to systematically target even the most subtle features in the solution that might be easily overlooked by a human.

Figure 4-17: Zoomed-in view of $\left| \frac{\partial^2 p}{\partial x \partial t} \right| \times LT$ with overlaid mesh
Figure 4-18: Second spatial derivative of adapted adjoint solution - $\left| \frac{\partial^2 \psi}{\partial x^2} \right| \times L^2$

Figure 4-19: Second time derivative of adapted adjoint solution - $\left| \frac{\partial^2 \psi}{\partial t^2} \right| \times T^2$
Figure 4-20: Second cross derivative of adapted adjoint solution - \( \left| \frac{\partial^2 \psi}{\partial x \partial t} \right| \times LT \)

Figure 4-21: Zoomed-in view of \( \left| \frac{\partial^2 \psi}{\partial x^2} \right| \times L^2 \) with overlaid mesh
Figure 4-22: Zoomed-in view of $|\frac{\partial^2 \psi}{\partial t^2}| \times T^2$ with overlaid mesh

Figure 4-23: Zoomed-in view of $|\frac{\partial^2 \psi}{\partial x \partial t}| \times LT$ with overlaid mesh
Chapter 5

Buckley-Leverett Equation

This chapter presents the results of applying the solution adaptive framework to the well-known Buckley-Leverett equation, which is widely used as a benchmark test problem for most reservoir simulators. Section 5.3 describes how the governing equations of the Buckley-Leverett problem can be recast into the space-time formulation.

5.1 Background

Introduced by Buckley and Leverett in 1942 [13], the Buckley-Leverett equation describes the mechanism by which one fluid is displaced by another in a porous medium. It is a nonlinear hyperbolic equation that contains a non-convex flux function, which implies that the solutions of this equation may contain shocks, rarefactions and even combinations of them. Although the vast amount of finite difference and finite volume literature on this problem will not be reviewed here, it is worth noting that there exist a few instances where finite element methods have been used. Juanes and Patzek have demonstrated solving the Buckley-Leverett problem using a continuous Galerkin finite element method stabilized using a multi-scale approach [32]. In [47], Riviére presents results for the Buckley-Leverett problem solved using a high-order time-marching DG method.
5.2 Governing equations

The incompressible two-phase flow equations in 1D can be combined and simplified into the nonlinear convection-diffusion scalar conservation law given below:

$$\phi \frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x} \left( u_T f_w(S_w) - D(S_w) \frac{\partial S_w}{\partial x} \right) = 0, \quad (5.1)$$

where the fractional flow function $f_w(S_w)$ and the diffusion coefficient arising from capillary effects $D(S_w)$ are given by:

$$f_w(S_w) = \frac{\lambda_w}{\lambda_w + \lambda_n}, \quad (5.2)$$

$$D(S_w) = -\frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n} \left( K \frac{\partial p_c}{\partial S_w} \right), \quad (5.3)$$

where $\lambda_w, \lambda_n$ are the relative mobilities,

$$\lambda_w = \frac{k_{rw}}{\mu_w} = \frac{S_w^r}{\mu_w}, \quad (5.4)$$

$$\lambda_n = \frac{k_{rn}}{\mu_n} = \frac{(1 - S_w)^r}{\mu_n}$$

$\phi$ is the porosity, $K$ is the absolute permeability, and $u_T$ is the total velocity. This work investigates problems with both quadratic relative permeability functions ($r = 2$) and quartic relative permeability functions ($r = 4$).

In the absence of capillary effects, the diffusion coefficient vanishes and Eq. (5.1) reduces to the hyperbolic Buckley-Leverett equation. But for most of the results presented below, the approach of Juanes and Patzek [32] has been followed to introduce a small amount of capillary effects and solve a near-hyperbolic problem. This was done to improve general solver robustness and ensure that the solution converges to the correct physical solution, especially for higher polynomial orders. A linear capillary pressure model was used so that $\frac{\partial p_c}{\partial S_w}$ was simply a negative constant.
5.3 Space-time formulation

As shown for the pressure equation in the previous chapter, the 1D unsteady saturation equation given by Eq. (5.1) can also be recast as a 2D “steady-state” conservation law, where the temporal dimension is treated like an additional spatial dimension. In particular, the space-time formulation of the saturation equation can be expressed as follows:

\[
\sum_{j=1}^{2} \frac{\partial}{\partial \tilde{x}_j} \mathcal{F}^{\text{inv}}_{j}(S_w) - \sum_{j=1}^{2} \frac{\partial}{\partial \tilde{x}_j} \mathcal{F}^{\text{visc}}_{j}(S_w, \nabla S_w) = 0, \tag{5.5}
\]

where,

\[
\mathcal{F}^{\text{inv}}(S_w) = \left( u_T f_w, \phi S_w \right) \quad \text{and} \quad \mathcal{F}^{\text{visc}}(S_w, \nabla S_w) = \left( D \frac{\partial S}{\partial x}, 0 \right) \tag{5.6}
\]

and the 2D “space-time” coordinate is given by \( \tilde{x} = [x, t] \).

The spatial component of the numerical flux at an interface is computed using Godunov’s flux for a general nonlinear conservation law [36], which ensures that the local Riemann problem at each interface is solved exactly. Nearly all P1 and P2 results presented in this section were obtained using a pseudo-time continuation method as described in Section 2.2.2. This was because all attempts at solving the steady discrete equation directly, using a line search and Newton’s method with P-sequecing, were unsuccessful. Although the concept of pseudo-time continuation is unclear when solving a space-time problem, it still introduced a form of relaxation to the problem that allowed the nonlinear solver to behave in a more robust manner.

5.4 Test problem

5.4.1 Problem statement

The test problem involves the displacement of oil by water in a one-dimensional reservoir of length \( L = 50 \text{ ft} \). Water enters the reservoir from the left end, which means
that the water saturation $S_w$ at the left boundary is held at unity. The governing equation is described by Eq. (5.1)-(5.4), with the initial condition:

$$S_w(x, 0) = 0 \quad \forall x \in [0, L], \quad (5.7)$$

and the following boundary conditions:

$$S_w(0, t) = 1 \quad \forall t \in [0, T] \quad (5.8)$$

$$S_w(L, t) = 0 \quad \forall t \in [0, T]$$

Other relevant parameters for the problem are:

$$\phi = 0.3, \quad u_T = 0.3 \text{ ft/day},$$

$$\mu_w = 1 \text{ cP}, \quad \mu_n = 2 \text{ cP},$$

$$K \frac{\partial p_c}{\partial S_w} = \begin{cases} 
0 \text{ cP ft}^2/\text{day,} & \text{hyperbolic equation (Buckley-Leverett)} \\
-0.1 \text{ cP ft}^2/\text{day,} & \text{nearly-hyperbolic equation}
\end{cases}$$

Figure 5-1 contains plots of the fractional flow function $f_w$ and the diffusion coefficient $D$ as functions of $S_w$, with quadratic and quartic relative permeability functions, for the parameters given above.

### 5.4.2 Objectives

The solution to the Buckley-Leverett equation is known to contain sharp discontinuities, i.e. saturation shocks, which propagate through the domain. A good numerical method needs to be able to capture the magnitude and speed of these discontinuities accurately. The numerical diffusion of first order methods such as finite volume methods cause the discontinuous saturation profile to diffuse or smear out, whereas high-order methods tend to produce spurious oscillations around the shock due to the Gibbs phenomenon. This work aims to utilize output-based mesh adaptation to mitigate the effects of both numerical diffusion and Gibbs phenomenon.
Figure 5-1: Functions for the Buckley-Leverett problem

The specific objectives are listed below:

- Accurately predict the profile of water saturation in the reservoir up to $T = 30$ days.

- Compare results with those obtained from a finite volume method.

Since the output functional of interest is typically a quantity of physical significance, it was initially chosen to be the integral of $S_w$ along the $t = T$ boundary of the space-time domain, which represents the total volume of water in the reservoir at time $t = T$, as given in Eq. (5.9).

$$J_0 = \int S_w(x, T) \, dx$$  \hfill (5.9)

However, it turns out that $J_0$ is not a suitable output for adaptation because it causes the adjoint/dual solution of the Buckley-Leverett equation (i.e. no capillary term) to be constant-valued over the whole space-time domain $\Omega$. This is shown in detail in Section A.1. Since the adjoint solution is constant, it can be represented exactly
using any piecewise polynomial space that is P0 or higher. This implies that the DWR error estimates calculated according to Eq. (2.33) become zero due to Galerkin orthogonality, thereby causing the adaptive process to fail. Furthermore, a time-marching FV method always produces the exact result for $J_0$, regardless of the cell size or timestep size (see Section A.3), rendering any attempts at comparing error convergence rates useless.

These issues led to the formulation of an alternative output functional $J_1$, which can be considered as a natural extension of $J_0$, to be the integral of $S_w$ over the whole space-time domain $\Omega$. $J_1$ can be expressed more formally as:

$$J_1 = \int_{\Omega} S_w(x, t) \, d\Omega$$

Unfortunately, $J_1$ also shares much of the same traits, with the corresponding adjoint solution turning out to be linear in time, as derived in Section A.2. A linear adjoint solution means that the true output could be determined using a piece-wise linear (P1) solution without requiring any adaptation, and any attempts at adaptation using P1 or higher-order elements fail due to the error estimates being zero. Nevertheless, the adaptive process works as expected with P0 elements, the results of which will be presented in the next section. However, since this work seeks to demonstrate the use of high-order adaptive methods, a better output functional was required. Towards this end, output functional $J_2$ was devised as given by Eq. (5.11), and although it lacks a useful physical meaning, it produces a non-trivial adjoint solution that can be used in the error estimation process to drive the adaptive method as expected, even for higher polynomial orders.

$$J_2 = \int_{\Omega} (S_w(x, t))^2 \, d\Omega$$

For this reason, $J_2$ is used as the output functional of interest for all high-order adaptive results presented in this chapter.
5.5 Numerical results

5.5.1 Hyperbolic equation

It is well known that the original Buckley-Leverett equation, i.e. in the absence of the capillary pressure term in Eq. (5.1), can be solved easily using a time-marching finite volume method, which can also be shown to be equivalent to a first-order (P0) time-marching DG scheme, when Backward Euler is used for time integration. The results presented in this section successfully demonstrate the use of a DG scheme with P0 elements to solve the hyperbolic Buckley-Leverett equation, both with a time-marching formulation and a general space-time formulation.

Figures 5-2 and 5-3 compare the final saturation profiles obtained using a first order time-marching DG method (solid blue line), a structured space-time DG method (solid black line) and an adaptive, fully unstructured space-time DG method that adapts to output functionals $J_1$ (solid magenta line) and $J_2$ (solid red line). All plots clearly show the existence of rarefaction-shocks; a well-known characteristic of Buckley-Leverett solutions. In both figures, the space-time DG solution adapted to $J_1$ contains the greatest amount of numerical dissipation, whereas the one adapted to $J_2$ is the closest to the analytical solution (dashed black line). The final adapted space-time meshes that produced the data for the magenta and red lines are given in Figures 5-4 and 5-5 for the quadratic relative permeability case, and in Figures 5-6 and 5-7 for the quartic case. After 20 adaptation iterations, all adapted meshes exhibit a large amount of mesh refinement around the saturation shock as expected, and contain very few elements to the right of shock since the constant saturation in that region can be captured exactly by P0 elements.
Figure 5-2: Saturation at $t = 30$ days, with quadratic $k_r$

Figure 5-3: Saturation at $t = 30$ days, with quartic $k_r$
Figure 5-4: Space-time mesh adapted to $J_1$ - P0 10k DOF with quadratic $k_r$

Figure 5-5: Space-time mesh adapted to $J_2$ - P0 10k DOF with quadratic $k_r$
Figure 5-6: Space-time mesh adapted to $J_1$ - P0 10k DOF with quartic $k_r$

Figure 5-7: Space-time mesh adapted to $J_2$ - P0 10k DOF with quartic $k_r$
Figure 5-8: Adjoint solution adapted to $J_1$ - P0 10k DOF with quadratic $k_r$

Figure 5-9: Adjoint solution adapted to $J_2$ - P0 10k DOF with quadratic $k_r$
The elements around the shock in Figures 5-4 and 5-6 appear to be, somewhat counter-intuitively, stretched more in the spatial direction, instead of being aligned with the shock. However, this is consistent with the corresponding adjoint solution shown in Figure 5-8, because a P0 solution requires more resolution in the temporal direction than in the spatial direction to capture the adjoint solution that is only a linear function of time.

The meshes adapted to output $J_2$ contain far less horizontal stretching around the shock compared to those adapted to $J_1$. Figures 5-5 and 5-7 show that the elements around the shock are well aligned along the shock, and also exhibit a significant amount of adaptation around a “bow”-like feature located slightly to the left of the saturation shock. From the adjoint solution given in Figure 5-9, it is clear that the “bow” is actually a sharp layer present in the adjoint solution, and therefore the adaptation focuses around it to minimize the adjoint error. The adjoint solutions of the quartic relative permeability case are very similar in profile to those given in Figures 5-8 and 5-9, and are therefore omitted.

Solving the Buckley-Leverett equation using a DG discretization that is greater than first order accurate (i.e. P1 elements or higher) proves to be more difficult, primarily due to the overshoots and undershoots that are caused by the Gibbs phenomenon near a discontinuity. These spurious oscillations cause the saturation profile to contain unphysical values that are below zero or above one, and as a consequence the method may even converge to an unphysical solution. Figure 5-10 gives an example of such a situation where the physically incorrect structured space-time DG result (solid black line) is actually a residual-converged solution. The shock speed of this incorrect solution is consistent with the negative values of saturation just after the shock. Although not visible in Figure 5-10, which shows the saturation profiles at $t = 30$, the undershoots in the space-time solution reach values of around -0.5 at intermediate times. This can be seen in the complete space-time solution in Figure 5-11.
Figure 5-10: Saturation at $t = 30$ days, with quadratic $k_r$ - P1 solutions

Figure 5-11: Space-time solution - P1 50x60 with quadratic $k_r$
The above observations imply that some form of solution limiters may be required to obtain the correct physical solution. However, the time-marching DG result with the larger timestep, $\Delta t = 0.25$, produces a solution that is closer to the analytical solution (solid blue line). Although this solution is suspiciously different from the space-time solution (solid black line), the time-marching solution eventually matches up with the space-time solution when the timestep size is reduced (solid red line). This is because the time-marching DG method uses a Backward Euler discretization for time, and hence requires a much smaller timestep to match the second order temporal accuracy of the space-time DG method. The larger amount of numerical dissipation present in the $\Delta t = 0.25$ case compared to the $\Delta t = 0.03$ case reduces the magnitude of undershoots and converges to a more physically correct solution.

All attempts at solving this hyperbolic equation using higher polynomial orders (i.e. P2 or higher) either converged to incorrect solutions or failed to converge at all. The general lack of solver robustness made it difficult to obtain higher order adaptive results. In order to overcome this issue, it was decided to introduce a small amount of diffusion into the Buckley-Leverett equation via capillary effects as given in Eq. (5.1). This choice was inspired by the effect of greater numerical dissipation on the time-marching DG solutions presented above, where diffusion helped stabilize any overshoots and undershoots, and allowed convergence to the desired physical solution.

5.5.2 Nearly-hyperbolic equation

Although the Buckley-Leverett equation was modified by adding a diffusion term to improve the robustness of the adaptive process, the magnitude of the capillary effects were minimized to keep the resulting equation nearly-hyperbolic. For clarity, this section only presents results for problems with quadratic relative permeability functions. It is anticipated that for problems with quartic relative permeability functions, the magnitude of $K \frac{\partial p}{\partial S_w}$ required to stabilize the undershoots and overshoots may be larger. This is primarily due to the difference in magnitude of the diffusion coefficient $D(S_w)$ between quadratic and quartic relative permeability functions, as seen in Figure 5-1b. All adaptive results presented in this section have been adapted to
the volume integral of the $S_w^2$ over the space-time domain, i.e. the output functional is $J_2$.

Figure 5-12 shows a piecewise linear (P1) adapted space-time solution containing approximately 30,000 degrees of freedom. All adaptive results presented in this chapter were obtained after 20 mesh adaptation iterations. It clearly shows the presence of a rarefaction-shock that is propagating to the right, across the spatial domain. Figure 5-13 shows the corresponding adjoint solution on the same adapted mesh. Although the adjoint solution also contains a prominent shock-like feature, it is not as sharp as the “bow”-like feature seen previously in the purely hyperbolic problem (Figure 5-9).

Figures 5-14 and 5-15 contain the final adapted space-time meshes for P1 and P2 solutions with a target DOF count of 30,000. The elements in the final adapted meshes seem well aligned with the features of the flow, especially around the saturation shock. The high-order P2 solution requires fewer elements than the P1 solution to capture the decaying saturation profile to the left of the shock, allowing it to focus more degrees of freedom around the shock. Since the constant solution in the blue region to the right of shock can be exactly captured by a solution of any polynomial order, both meshes only contain a handful of elements in that region, as expected.
Figure 5-12: Adapted space-time primal solution - P1 30k

Figure 5-13: Adapted space-time adjoint solution - P1 30k
Figure 5-14: Adapted space-time mesh - P1 30k

Figure 5-15: Adapted space-time mesh - P2 30k
5.5.3 Comparisons with a conventional method

The aim of this section is to provide comparisons between the adapted space-time DG results and those obtained from a conventional time-marching finite volume method. For this purpose, all results were obtained by solving the nearly-hyperbolic equation (i.e. with small capillary effects) using quadratic relative permeability functions.

Figure 5-16 compares the saturation profiles at $t = 30$ obtained by the different methods. All three saturation profiles in the figure appear virtually identical and agree well with each other.

A more rigorous analysis was also carried out to compare the performance and efficiency of each numerical method in solving this problem. Figure 5-17 shows how the output error, $\mathcal{E} \equiv J_2 - J_{2h,p}$, behaves with the average size of an element in the mesh for three different numerical methods: time-marching FV, space-time DG with uniformly refined structured meshes, and space-time DG with output-based adaptation. Since the true output $J_2$ was not known analytically, it was estimated by using a space-time adapted DG solution containing around 100,000 degrees of freedom. In order to cut down on the large computation time, first a 30k DOF P1 solution was obtained by mesh adaptation, and then the problem was re-solved on the
final adapted mesh using P3 elements. The numbers next to each data point on the figure represent the total number of space-time DOF in that solution, except for the FV results where the numbers show the number of spatial cells used in the simulation. The average element size $h$ for each case is estimated as described previously in Eq. (4.10).

The dashed blue line representing time-marching FV results has a slope close to 1, exhibiting first order convergence as expected. Above a certain mesh resolution, the space-time DG results obtained using uniformly refined structured meshes start to outperform the FV results, with the P1 and P2 structured mesh results having slopes of around 1.2 and 3 respectively. The adaptive space-time DG method performs exceptionally well, and is the most efficient out of the three methods, in terms of error-per-DOF. The high-order adapted results achieve errors that are 2 to 4 orders of magnitude smaller than the best results on structured meshes, and with far fewer

![Figure 5-17: Error vs. $h$](image-url)

The dashed blue line representing time-marching FV results has a slope close to 1, exhibiting first order convergence as expected. Above a certain mesh resolution, the space-time DG results obtained using uniformly refined structured meshes start to outperform the FV results, with the P1 and P2 structured mesh results having slopes of around 1.2 and 3 respectively. The adaptive space-time DG method performs exceptionally well, and is the most efficient out of the three methods, in terms of error-per-DOF. The high-order adapted results achieve errors that are 2 to 4 orders of magnitude smaller than the best results on structured meshes, and with far fewer
degrees of freedom.

An attempt was also made to compare the runtime efficiency of the different methods, which may be regarded by some to be of greater practical importance than the DOF efficiency discussed above. Unfortunately, the finite volume method was omitted in the analysis because the FV method was coded in MATLAB, and the space-time DG method was coded in C, and comparing execution times of codes written in different languages does not yield a fair judgement about each algorithm’s efficiency. Figure 5-18 shows the total time (in seconds) taken by the space-time DG method on structured meshes and the adaptive space-time DG method, to achieve an output error $\mathcal{E}$. Note that the runtime of an adaptive result includes the time taken to generate multiple intermediate meshes, compute solutions, and also perform error estimation. All simulations were run on a single thread of an Intel Core i7-3630QM (2.4GHz) processor, with a cache size of 6MB and 16GB of RAM.

Not surprisingly, the adaptive space-time method dominates the runtime efficiency results too. For a given amount of time, the adaptive space-time method is able to consistently produce results with errors that are at least 2 orders of magnitude smaller than those obtained by solving on uniformly refined structured meshes.

5.6 Summary

This chapter has presented a space-time DG formulation for a general form of the Buckley-Leverett equation which includes capillary effects. The results presented demonstrate the successful application of the adaptive framework to a test problem. In the absence of capillary pressure, the method is robust when P0 elements are used, i.e. for a first-order discretization. Using P1 or higher order elements produces spurious oscillations due to Gibbs phenomenon, which in turn modifies shock speeds and causes the method to converge to physically incorrect solutions. Introducing a small amount of diffusion in the form of capillary effects helps mitigate this issue, and the adaptive method is then able to robustly converge to the correct solution. Comparisons with a conventional time-marching FV method and also with a space-time
DG method on uniformly refined structured meshes show that the fully unstructured adaptive space-time DG method is vastly superior in terms of DOF efficiency and runtime efficiency.
Chapter 6

Two-Phase Flow

This chapter introduces the two-phase flow equations in mass conservation form and presents the results of applying the solution adaptive framework to an oil-water test problem. Section 6.3 describes how the governing equations of the single-phase flow problem can be recast into the space-time formulation.

6.1 Background

The existing literature on the use of high-order discontinuous Galerkin finite element methods for solving two-phase flow problems is quite sparse, and almost all of the existing work use a decoupled formulation where a discrete pressure equation and a discrete saturation equation are solved sequentially. This technique, also known as the IMPES (implicit pressure, explicit saturation) method is applicable for incompressible flows, and is particularly useful since it allows the coupled elliptic-hyperbolic system of equations to be separated into a purely elliptic “pressure” equation and a hyperbolic “saturation” equation. Additional details of the IMPES method can be found in [4].

The pressure equation can then be solved using a primal DG formulation for diffusion problems, such as the Oden-Baumann-Babuska (OBB) method [40] used by Klieber in [33]. Both equations can also be discretized using the non-symmetric interior penalty Galerkin method (NIPG) [50], symmetric interior penalty Galerkin method (SIPG) [3, 54] or the incomplete interior penalty Galerkin method (IIPG)
as demonstrated by Epshteyn in [26, 24]. However in this work, the sequential approach is abandoned and the coupled system of equations is solved simultaneously in a fully implicit manner. Although this is comparable to the work of Epshteyn in [25], the method presented in this thesis uses the BR2 formulation [8, 9] to discretize the diffusion terms, instead of the class of DG methods that require a choice of a penalty parameter (NIPG, SIPG, IIPG).

To the best of the author’s knowledge, this work is the first attempt at solving a two-phase flow problem using an adaptive high-order DG method in a coupled, fully implicit manner. Furthermore, the use of fully unstructured space-time meshes only adds to the novelty of this work.

6.2 Governing equations

The Darcy velocities of the wetting (w) and non-wetting (n) phases in a two-phase flow can be expressed as follows, using the general form of Darcy’s law given in Eq. (3.7):

\[ u_w = -\frac{K_{rw}}{\mu_w} (\nabla p_w - \rho_w g) \]

\[ u_n = -\frac{K_{rn}}{\mu_n} (\nabla p_n - \rho_n g) \]

These velocities can then be used to derive a statement of mass conservation for each of the two phases in the porous medium:

\[ \frac{\partial (\rho_w \phi S_w)}{\partial t} + \nabla \cdot (\rho_w u_w) = \rho_w q_w \]

\[ \frac{\partial (\rho_n \phi S_n)}{\partial t} + \nabla \cdot (\rho_n u_n) = \rho_n q_n \]

where \( q_w \) and \( q_n \) represent source/sink terms for the wetting and non-wetting phases respectively, and are specified as volumetric flow rates per unit volume (positive \( q \) indicates mass injection). Furthermore, the system of equations also require the
following closure relation for the phase saturations:

\[ S_w + S_n = 1 \]  (6.3)

For the work in this thesis, Eq. (6.4) is further simplified using the following assumptions:

- Negligible gravitational effects
- Constant fluid and rock compressibilities
- Constant permeability field \( K \)
- Constant fluid viscosities \( \mu_w, \mu_n \)

After substituting the Darcy velocity expressions into Eq. (6.2), and recognizing that the fluid densities \( \rho_w, \rho_n \) and porosity \( \phi \) are explicit functions of the phase pressures, it appears as if there are four unknown variables to solve for: \( p_w, p_n, S_w \) and \( S_n \). But thanks to the closure relation given in Eq. (6.3) and the definition of a capillary pressure \( p_c(S_w) = p_n - p_w \), the system of equations can be expressed in terms of only two unknowns.

The choice of the two primary variables is somewhat arbitrary, although it is mostly formulated with one pressure variable and one saturation variable. The wetting phase pressure \( (p_w) \) and the wetting phase saturation \( (S_w) \) is used by Rivière in [46], whereas Epshteyn uses \( p_w, S_n \) as the primary variables in [24, 25]. It is also possible to formulate the problem in terms of a global pressure and \( S_n \), as is done in [26], where the global pressure is defined according to the work of Chavent and Jaffré [15]. This work uses the \( p_w, S_w \) formulation for the two-phase flow equations, which under the assumptions stated above, can be written as:

\[
\frac{\partial}{\partial t} (\rho_w \phi S_w) - \nabla \cdot \left( \rho_w \frac{K}{\mu_w} \nabla p_w \right) = \rho_w q_w 
\]  (6.4)

\[
\frac{\partial}{\partial t} (\rho_n \phi (1 - S_w)) - \nabla \cdot \left( \rho_n \frac{K}{\mu_n} \nabla p_w + \frac{\partial p_c}{\partial S_w} \nabla S_w \right) = \rho_n q_n
\]
Note that the spatial derivative of the capillary pressure in the second equation has been expanded out using the chain rule: \( \nabla p_c = \sigma_{pw} \frac{\partial p_w}{\partial S_w} \nabla S_w \).

### 6.3 Space-time formulation

By recasting the \( d \)-dimensional unsteady conservation law in Eq. (6.4) as a \((d + 1)\)-dimensional “steady-state” conservation law, the space-time formulation of the two-phase flow equations can be given as follows:

\[
\sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} F_{i}^{\text{inv}}(u) - \sum_{j=1}^{d+1} \frac{\partial}{\partial \hat{x}_j} F_{i}^{\text{visc}}(u, \nabla u) = S(u, \hat{x}), \tag{6.5}
\]

where \( u = [p_w, S_w]^T \) and the augmented space-time coordinate is \( \hat{x} = [x, t] \). The fluxes are given by:

\[
F_{i}^{\text{inv}}(u) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{for } i = 1, \ldots, d \tag{6.6}
\]

\[
F_{d+1}^{\text{inv}}(u) = \begin{pmatrix} \rho_w \phi S_w \\ \rho_n \phi (1 - S_w) \end{pmatrix} \tag{6.7}
\]

\[
F_{i}^{\text{visc}}(u, \nabla u) = \begin{pmatrix} \rho_w \frac{k_{nw}}{\mu_w} K_{ij} \phi \frac{\partial p_w}{\partial \hat{x}_j} \\ \rho_n \frac{k_{nn}}{\mu_n} K_{ij} \phi \left( \frac{\partial p_w}{\partial \hat{x}_j} + \frac{\partial p_n}{\partial S_w} \frac{\partial S_w}{\partial \hat{x}_j} \right) \end{pmatrix} \quad \text{for } i = 1, \ldots, d \tag{6.8}
\]

\[
F_{d+1}^{\text{visc}}(u, \nabla u) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{6.9}
\]

\[
S(u, \hat{x}) = \begin{pmatrix} \rho_w q_w \\ \rho_n q_n \end{pmatrix} \tag{6.10}
\]

It is worth mentioning that all high-order solutions presented for this problem were solved using the P-sequencing continuation method described in Section 2.2.2. However, all sub-problems were started from P1 instead of P0, since the BR2 method does not yield a consistent discretization of the viscous fluxes \( F_{i}^{\text{visc}} \) with P0 elements.
6.4 Test problem

6.4.1 Problem statement

The two-phase flow test problem involves a one-dimensional reservoir from which oil is extracted by water injection as shown in the schematic in Figure 6-1. The length of the reservoir is given by $L = 1000$ ft, and it is assumed to be initially completely filled with oil at a constant pressure of 2500 psi. An injection well of length $L_s = 10$ ft injects water at a constant rate of $Q = 100$ barrels per day, and an extraction well of equal length extracts fluid at the same rate. The reservoir is assumed to have a constant width $\Delta y$ and constant depth $\Delta z$, so that the injection and extraction rates can be specified as volumetric flow rates, as in a real reservoir. In this setup, the wetting phase ($w$) is water and the non-wetting phase ($n$) is oil.

![Schematic of 1D reservoir](image)

Figure 6-1: Schematic of 1D reservoir

The governing equations are given by Eq. (6.4) with the initial condition:

$$p_w(x, 0) = 2500 \text{ psi} \quad \forall x \in [0, L]$$  \hspace{1cm} (6.11)

$$S_w(x, 0) = 0,$$  \hspace{1cm} (6.12)

and no flow boundary conditions at the left and right ends of the reservoir, which reduce to the following Neumann conditions:

$$\nabla u(0, t) = \nabla u(L, t) = 0 \quad \forall t \in [0, T].$$  \hspace{1cm} (6.13)
Other relevant relationships for densities, porosity, relative permeabilities and capillary pressure are given below:

\[
\rho_\alpha = \rho_{\alpha\text{ref}} e^{c_\alpha(p_\alpha - p_{\text{ref}})} \quad \forall \alpha \in \{w, n\}
\]

\[
\phi = \phi_{\text{ref}} e^{c_\phi(p_n - p_{\text{ref}})}
\]

\[
k_{rw} = S^2_w
\]

\[
k_{rn} = S^2_n = (1 - S_w)^2
\]

\[
p_c = p_{c_{\text{max}}}(1 - S_w)
\]

where,

\[
\phi_{\text{ref}} = 0.3, \quad p_{\text{ref}} = 14.7 \text{ psi},
\]

\[
\rho_{w\text{ref}} = 62.4 \text{ lbf/ft}^3, \quad \rho_{n\text{ref}} = 52.1 \text{ lbf/ft}^3,
\]

\[
c_w = 5 \times 10^{-6} \text{ psi}^{-1}, \quad c_n = 1.5 \times 10^{-5} \text{ psi}^{-1},
\]

\[
c_\phi = 3 \times 10^{-6} \text{ psi}^{-1}, \quad K = 200 \text{ md},
\]

\[
\mu_w = 1 \text{ cP}, \quad \mu_n = 2 \text{ cP},
\]

\[
Q = 100 \text{ bbls/day}, \quad L_s = 10 \text{ ft},
\]

\[
\Delta y = 100 \text{ ft}, \quad \Delta z = 30 \text{ ft},
\]

\[
p_{c_{\text{max}}} = 1 \text{ psi}.
\]

The wells are modeled by the discontinuous source terms \(q_w(x, t)\) and \(q_n(x, t)\) which are defined as follows:

\[
q_w = \begin{cases} 
\frac{Q}{L_s \Delta y \Delta z} & \text{for } 0 \leq x \leq L_s \\
-\frac{\lambda_w}{\lambda_w + \lambda_n} \frac{Q}{L_s \Delta y \Delta z} & \text{for } (L_s - L_s) \leq x \leq L \\
0 & \text{otherwise}
\end{cases}
\]

\[(6.14)\]
\[ q_n = \begin{cases} 
0 & \text{for } 0 \leq x \leq L_s \\
-\frac{\lambda_n}{\lambda_w + \lambda_n} \frac{Q}{L_s \Delta y \Delta z} & \text{for } (L - L_s) \leq x \leq L \\
0 & \text{otherwise}
\end{cases} \quad (6.15) \]

where the relative mobilities \( \lambda_w, \lambda_n \) are given by:

\[ \lambda_\alpha = \frac{k_\alpha}{\mu_\alpha} \quad \forall \alpha \in \{w, n\} \quad (6.16) \]

The volumetric flow rate of each phase at the extraction well depends on the saturation of the fluid through the relative mobilities, which implies that the source terms on the right-hand side of Eq. (6.4) are solution dependent. Note that appropriate unit conversions were also applied to the quantities given above to make the equations dimensionally consistent.

### 6.4.2 Objectives

The solution of this two-phase flow problem is expected to contain a combination of the features observed in the solutions of previous test problems. Therefore the objective of this test problem is to investigate how well the adaptive method can capture the irregular features of the saturation variable, which is also tightly coupled to a relatively smooth pressure variable. Unfortunately, the adaptive results could not be compared to those from a conventional time-marching FV method like in previous chapters, due to time constraints. The specific objectives of this test problem are listed below:

- Accurately predict the profile of water saturation in the reservoir up to \( T = 100 \) days.
- Compare results obtained using the space-time adaptive method against those obtained on uniformly refined structured meshes without output-based adaptation.
For this purpose, the output of interest as defined in Eq. (2.25) was chosen to be the integral of $S_w^2$ over the entire space-time domain:

$$J = \int_{\Omega} (S_w(x,t))^2 \, d\Omega$$ \hspace{1cm} (6.17)

6.5 Numerical results

6.5.1 Adapted solutions

Figures 6-2 and 6-3 show the wetting phase pressure and saturations obtained using a piecewise quadratic (P2) adapted space-time solution containing approximately 30,000 degrees of freedom (DOF) for each primary variable. All adaptive results presented in this chapter were obtained after 20 mesh adaptation iterations. The pressure $p_w$ starts with a constant value of 2500 psi at $t = 0$ and then quickly transitions into a nearly linear profile. The negative spatial gradient of the pressure is set by the source terms which increase the pressure at the left end by water injection, and decrease pressure at the right end by fluid extraction. The initial pressure transient observed along the bottom boundary of the space-time domain is due to the parabolic nature of $p_w$, which is a result of the equations being slightly compressible. In contrast, the plot of $S_w$ in Figure 6-3 clearly shows a rarefaction-shock propagating to the right at a constant speed (after the initial pressure transient). Note that this plot has been zoomed in to show only the initial 150 ft of the domain for clarity. The value of $S_w$ in the rest of the domain is zero.

Figures 6-4 and 6-5 contain plots of the adjoint variables, $\psi_w$ and $\psi_n$, respectively. $\psi_n$ in particular contains a sharp layer that is aligned in the same direction as the saturation shock, but is located slightly to the left of it. The final adapted meshes for the P1 and P2 cases are given in Figures 6-6 and 6-7. Both meshes show significant adaptation near the left boundary due to the presence of the injection well and the saturation shock, and also near the right boundary due to the extraction well. The center of the space-time domain contains very few elements since both P1 and P2 solutions can accurately represent the linear pressure and constant saturation profiles.
in that region. The horizontal clustering of elements near the bottom boundary, particularly in the P1 mesh, is required to accurately capture the sharp temporal gradients in the initial pressure transient. Figures 6-8 and 6-9 contain closer views of the mesh around the saturation shock, which show significant adaptation occurring along the saturation shock, and also along the $x = 10$ ft line where the source term ends.

The P1 solution requires a larger amount of elements than the P2 solution to represent the saturation rarefaction to the left of the shock, which is as expected. Since the adaptive method strives to reduce both primal and dual errors, the adaptation process also focuses around the sharp layer in $\psi_n$ which is located slightly to the left of the shock, as seen in Figure 6-8.
Figure 6-2: Adapted space-time solution - P2 30k DOF - pressure ($p_w$)

Figure 6-3: Adapted space-time solution - P2 30k DOF - saturation ($S_w$)
Figure 6-4: Adapted space-time solution - P2 30k DOF - adjoint $\psi_w$

Figure 6-5: Adapted space-time solution - P2 30k DOF - adjoint $\psi_n$
Figure 6-6: Adapted space-time mesh - P1 30k DOF - full domain

Figure 6-7: Adapted space-time mesh - P2 30k DOF - full domain
Figure 6-8: Adapted space-time mesh - P1 30k DOF - initial 150 ft

Figure 6-9: Adapted space-time mesh - P2 30k DOF - initial 150 ft
6.5.2 Comparisons with uniform refinement

This section compares the results from the space-time adaptive DG method with those obtained by a space-time DG method on uniformly refined structured meshes. It should be mentioned that the initial structured mesh is not perfectly uniform, but consists of several layers of elements as shown in Figure 6-10. This was done to avoid having to refine the mesh an infeasible number of times before obtaining any reasonable solutions that are worth comparing with the adaptive solutions. Although the generation of such a structured mesh requires some \textit{a priori} knowledge of the solution, and thereby biases the comparison, the results soon show that this bias is irrelevant.

![Initial layered structured mesh - 30 × 20](image)

Figure 6-10: Initial layered structured mesh - 30 × 20

Figure 6-11 shows the water saturation profiles at $t = 100$ days, as predicted by the space-time DG method with adapted and uniformly refined structured meshes. All three results agree well with each other for most of the domain, except just around the saturation shock. As expected, the P2 solution on the structured mesh (solid black line) has large oscillations around the shock, whereas the adapted P1 and P2
solutions have minimal undershoots.

Figure 6-11: $S_w$ at $t = 100$ days

A more careful comparison between the adapted and structured DG results was carried out, the results of which are summarised in Figures 6-12 and 6-13. Figure 6-12 compares how the output error, $\mathcal{E} \equiv J - J_{h.p}$, behaves with the average size of an element for the space-time DG method on uniformly refined structured meshes, and the space-time DG method with output-based adaptation. Since the true output $J$ was not known analytically, it was estimated by using a space-time adapted DG solution containing over 80,000 degrees of freedom for each primary variable. In order to cut down on the large computation time, first a 50k DOF P2 solution was obtained by mesh adaptation, and then the problem was re-solved on the final adapted mesh using P3 elements. The numbers next to each data point on the figure represent the total number of space-time DOF in that solution, and the average element size $h$ for each case is estimated as described previously in Eq. (4.10).

The adapted results show a clear distinction from those obtained on structured meshes by consistently achieving error levels that are at least 2 orders of magnitude smaller for a given $h$ value. The P1 and P2 structured mesh results have slopes of around 1.4 and 3.4 respectively, whereas the adapted results are observed to have
Figure 6-12: Error vs. $h$

Figure 6-13: Error vs. Runtime
slightly higher convergence rates of 2.7 and 4.2, for P1 and P2 respectively.

Figure 6-13 shows the total time (in seconds) taken by each method to achieve an output error $\mathcal{E}$. Note that the runtime of the adaptive results include the time taken to generate multiple intermediate meshes, compute solutions, and also perform error estimation. All simulations were run on a single thread of an Intel Core i7-3630QM (2.4GHz) processor, with a cache size of 6MB and 16GB of RAM. As expected, output-based adaptation is observed to be more runtime-efficient than uniform refinement for this problem, as it is able to produce more accurate outputs within a given amount of computation time.

### 6.5.3 Issues with more nonlinearity

The space-time adaptive method was also tried on a problem with more nonlinearity. Specifically, the relative permeability models $k_{rw}$ and $k_{rn}$ were chosen to be quartic functions instead of quadratic, and all other problem parameters were left unchanged. Since all attempts at solving this modified problem failed within the first couple of adaptation iterations, it was decided to increase the amount of capillary pressure in the system, by increasing $p_{cmax}$ to 5 psi. The additional diffusion introduced by increasing capillarity enabled adaptation to continue for longer, but it still failed eventually. The adaptive process failed because the update fraction $\eta$ of the line search in the nonlinear solver kept cutting back until $\eta < \eta_{\text{min}}$. This suggests that the Newton solver is probably stuck at a local minimum where it cannot find a suitable direction to move, without increasing the residual norms.

Figures 6-14 to 6-17 show the pressure and saturation solutions obtained after 8 adaptation iterations, which is the last residual-converged solution before the nonlinear solver stalled during the 9th primal solve. The sharp change in pressure gradient across the saturation shock is more prominent for this problem compared to the problem with quadratic relative permeability, and the difference is clearly visible in Figure 6-14. Figure 6-15 shows that the jump in saturation at the shock is larger for this problem, which also causes the saturation undershoots to be more negative as seen in Figure 6-17.
Figure 6-14: Comparison of $p_w$ profiles at $t = 100$ days from the two problems

Figure 6-15: Comparison of $S_w$ profiles at $t = 100$ days from the two problems
Figure 6-16: Adapted space-time solution for quartic $k_r$ case - P1 30k DOF - $p_w$

Figure 6-17: Adapted space-time solution for quartic $k_r$ case - P1 30k DOF - $S_w$
Figure 6-18: Adapted space-time mesh for quartic $k_r$ case - P1 30k DOF

Figure 6-19: Adapted space-time mesh for quartic $k_r$ case - P1 30k DOF - first 150 ft
Figure 6-20: Adapted space-time solution for quartic $k_r$ case - P1 30k DOF - $\psi_w$

Figure 6-21: Adapted space-time solution for quartic $k_r$ case - P1 30k DOF - $\psi_n$
Figures 6-18 and 6-19 show full and zoomed-in views of the adapted space-time mesh after the 8th iteration. The adapted meshes look reasonable, containing a significant amount of adaptation around the saturation shock and at the edge of the source terms as expected. However, the mesh at the center of the domain can be coarsened further, as seen previously for the quadratic $k_r$ case in Figure 6-6, which suggests that the adaptive process is incomplete.

The reason for the failure of the nonlinear solver and consequently also the adaptive process is not yet fully understood. However, it is speculated that the cause lies in the manner in which this advection-dominant system of equations is discretized. The rest of this section aims to discuss the reasoning behind this speculation.

One of the key points in the arguments that follow is that the two-phase flow equations can give rise to an advection-dominant flow, despite the fact that the governing equations given in Eq. (6.4) contain diffusive fluxes. Recall that by assuming incompressibility, the 1D two-phase flow equations can be simplified and re-arranged to form an elliptic pressure equation and the nonlinear convection-diffusion saturation equation given in Eq. (5.1). In the limit of zero capillary pressure, the convection-diffusion saturation equation becomes a pure convection equation (i.e. the Buckley-Leverett equation) while the pressure equation retains its elliptic nature.

It is believed that the failure mode arises from the manner in which the two-phase flow equations are discretized in the space-time DG framework. As described in Section 6.3, all of the spatial fluxes in the system of equations are interpreted as viscous fluxes and are therefore discretized using the BR2 method described in Section 2.1.2. Considering that the BR2 discretization is a central-difference operator that has no upwind-bias, stability issues are to be expected if the flow is advection dominant.

The lesser nonlinearity of the quadratic $k_r$ case may have allowed the BR2 discretization to survive with $p_{cmax} = 1$ psi, but the quartic $k_r$ case runs into convergence issues even with $p_{cmax} = 5$ psi. Although it is known that increasing the amount of diffusion via capillary effects will eventually provide enough stability for the BR2
discretization to work properly, this approach may require unphysically large values of $p_{c_{\text{max}}}$. In order to avoid this, it is expected that an appropriate modification of the discretization is necessary, and a few initial thoughts in this direction are discussed in the final chapter.

6.6 Summary

This chapter has presented a space-time DG formulation for the compressible two-phase flow equations in mass-conservation form. The equations are solved in a fully coupled manner, with the wetting phase pressure and saturation as the primary variables. The space-time adaptive DG method is demonstrated on a simple test problem, where the space-time mesh is adapted to an output functional that depends on the saturation. Solution features such as pressure transients and saturation shocks are captured efficiently by the adaptive method, and formal comparisons with results obtained on uniformly refined structured meshes verify that output-based adaptation is superior in terms of both DOF efficiency and runtime efficiency. The latter part of the chapter presents results from an incomplete adaptive simulation which was attempted on a highly nonlinear test problem. The section concludes with a discussion of the possible causes of failure of the nonlinear solver.
Chapter 7

Conclusion

7.1 Summary

This work aims to develop an autonomous, efficient and robust numerical method for solving reservoir flow problems, by attempting to unify three distinct approaches: high-order methods, solution adaptive methods, and space-time methods. A space-time discontinuous Galerkin method was coupled with the DWR method [10, 11] for obtaining output error estimates, which were then used by the MOESS algorithm [55] to perform mesh adaptation.

The proposed solution adaptive framework was applied to three standard reservoir flow problems of increasing complexity: the single-phase flow problem, the Buckley-Leverett problem, and the two-phase flow problem. Space-time DG formulations were presented for each problem, and the space-time adaptive DG method was successfully demonstrated on all three problems. Comparisons of the results from the proposed method with those from other methods, such as the time-marching FV method, showed that the space-time adaptive method was generally much more efficient in terms of both DOF and runtime, except in the case of smooth, linear problems (such as the single phase problem) for which a high-order method alone (without adaptation) proved to be the most efficient.

The work in this thesis can be identified as a clear indicator for the significant potential of high-order, solution adaptive numerical methods, especially with regards
to applications in porous media flows. It is hoped that this initial application will be improved upon and extended to more realistic, more complex flow problems in the future.

7.2 Future work

This section contains a few brief discussions on possible areas for further research, including a few useful remarks and insight obtained from this work.

Heterogeneous porous media

A common assumption throughout this thesis was the presence of a homogenous porous medium, i.e. rock properties such as porosity $\phi$ and permeability $K$ were constant, which is unfortunately rarely the case in real world applications. Since geological features such as faults can have a significant impact on the flow solution, current CFD practitioners need to identify such features manually and modify the mesh/discretization appropriately. Alternatively, one could allow a solution adaptive numerical method to identify such features autonomously. Considering the efficiency gains observed in this work due to adaptation around sharp features like shocks, it can be expected that an output-based adaptation algorithm will perform even better in the presence of prominent geological features.

Additional spatial dimensions

All the problems considered in this thesis can be extended to two spatial dimensions, which means that the corresponding space-time meshes will be three-dimensional. Although this would require a three-dimensional metric based meshing software to generate new meshes at adaptation iteration, the mostly box-shaped space-time domains should simplify the task. However, solving 3D spatial problems in space-time may be significantly more difficult, since it requires the generation of 4D meshes, which is still an active area of research.
Shock-capturing

The results of the Buckley-Leverett and two-phase flow problems have clearly shown the presence of discontinuous saturation profiles, which as discussed previously, cause spurious oscillations due to Gibbs phenomenon when discretized using high-order methods. Furthermore, unphysical negative undershoots were observed to yield incorrect solutions; a phenomenon which was mitigated by introducing a small amount of diffusive capillary effects together with mesh adaptation. A far better solution would be to implement a shock-capturing method as a part of the discretization. Conceptually, a shock-capturing method uses the same idea of introducing (artificial) diffusion to smooth out oscillations near shocks, but is capable of doing so in an intelligent manner by introducing diffusion only where it is needed, rather than globally as in Section 5.5.2.

The latter part of Chapter 6 presented a brief discussion of why the BR2 discretization of viscous fluxes used in this work may not be sufficient for the advection-dominant two-phase flow equations. A possible remedy for this issue might be to introduce an additional term into the DG discretization which effectively upwinds or biases the central difference BR2 operator appropriately. Some preliminary work done towards this end for an incompressible flow suggests that this term is most likely an interface flux term that is a function of the jump in saturation across the interface.
Appendix A

Adjoint Analysis of the Buckley-Leverett Equation

The Buckley-Leverett problem can be written as follows:

\[
\frac{\partial}{\partial t} (\phi S_w) + \frac{\partial}{\partial x} (u_T f(S_w)) = 0 \quad (A.1)
\]

\[
S_w(0, t) = 1 \quad t \in [0, T] \quad (A.2)
\]

\[
S_w(x, 0) = 0 \quad x \in [0, L] \quad (A.2)
\]

where \( \phi \) and \( u_T \) are constants and the fractional flow function \( f(S_w) \) is given by:

\[
f(S_w) = \frac{\lambda_w}{\lambda_w + \lambda_n} \quad (A.3)
\]

where \( \lambda_w = \frac{k_w}{\mu_w} \), \( \lambda_n = \frac{k_n}{\mu_n} \) are the relative mobilities.

This 1D unsteady equation can be re-cast as a 2D “steady” equation in space-time:

\[
\nabla \cdot \mathbf{F} = 0 \quad (A.4)
\]

where

\[
\mathbf{F} = \begin{bmatrix} u_T f(S_w) \\ \phi S_w \end{bmatrix} \quad (A.5)
\]
The following analysis assumes that the final time $T$ is chosen so that the saturation shock has not reached the right boundary. In other words, the following condition is assumed:

\[ S_w(L, t) = 0 \quad t \in [0, T] \quad (A.6) \]

**A.1 Output $J_0$: boundary integral of $S_w$**

This section presents the adjoint solution for the case where the output functional is the integral of $S_w$ along the $t = T$ boundary of the space-time domain $\Omega$.

\[ J_0 = \int_0^L S_w(x, T) \, dx \quad (A.7) \]

Integrating Eq. (A.4) over the space-time domain $\Omega$ and applying the divergence theorem:

\[ \int_\Omega \nabla \cdot F \, d\Omega = 0 \quad (A.8) \]
\[ \int_{\partial \Omega} F \cdot n \, dS = 0 \quad (A.9) \]

Expanding out boundary integrals:

\[ -\int_0^L \phi S_w(x, 0) \, dx + \int_0^L \phi S_w(x, T) \, dx - \int_0^T u_T f(S_w(0, t)) \, dt + \int_0^T u_T f(S_w(L, t)) \, dt = 0 \quad (A.10) \]

\[ -0 + \phi J_0 - \int_0^T u_T f(1) \, dt + \int_0^T u_T f(0) \, dt = 0 \quad (A.11) \]

Noting that $f(0) = 0$ and $f(1) = 1$:

\[ J_0 = \int_0^T \frac{u_T}{\phi} \, dt = \frac{u_T}{\phi} T \quad (A.12) \]
Therefore the chosen output $J_0$ is constant.

Derive the adjoint equation by considering infinitesimal perturbations to $S_w$, i.e. $\tilde{S}_w$:

\[
\tilde{J}_0 = \int_\Omega \psi \nabla \cdot \tilde{F} \, d\Omega
\]  
(A.13)

\[
\int_0^L \tilde{S}_w(x, T) \, dx = \int_{\partial \Omega} \psi \tilde{F} \cdot \mathbf{n} \, dA - \int_\Omega \nabla \psi \cdot \tilde{F} \, d\Omega
\]  
(A.14)

Adjoint equation:

\[
\nabla \psi \cdot \tilde{F} = 0  
\]  
(A.15)

\[
\phi \tilde{S}_w \frac{\partial \psi}{\partial t} + u_T f(\tilde{S}_w) \frac{\partial \psi}{\partial x} = 0  
\]  
(A.16)

Adjoint BCs:

\[
\int_0^L \tilde{S}_w(x, T) \, dx = -\int_0^L \psi(x, 0) \cdot \phi \tilde{S}_w(x, 0) \, dx + \int_0^L \psi(x, T) \cdot \phi \tilde{S}_w(x, T) \, dx 
\]

\[
- \int_0^T \psi(0, t) \cdot u_T f(\tilde{S}_w(0, t)) \, dt + \int_0^T \psi(L, t) \cdot u_T f(\tilde{S}_w(L, t)) \, dt
\]  
(A.17)

Noting that the perturbations $\tilde{S}_w = 0$ on the left boundary (i.e. $\tilde{S}(0, t) = 0$) and the bottom boundary (i.e. $\tilde{S}(x, 0) = 0$), and that $f(0) = 0$:

\[
\int_0^L \tilde{S}_w(x, T) [1 - \phi \psi(x, T)] \, dx = -\int_0^T \psi(0, t) \cdot u_T f(0) \, dt + \int_0^T \psi(L, t) \cdot u_T f(0) \, dt 
\]

\[
= 0
\]

\[
\Rightarrow 1 - \phi \psi(x, T) = 0
\]

\[
\psi(x, T) = \frac{1}{\phi}
\]  
(A.18)

The constant adjoint solution $\psi(x, t) = \frac{1}{\phi}$ satisfies Eq. (A.16) and the adjoint boundary condition in Eq. (A.18).
A.2 Output $J_1$: volume integral of $S_w$

This section presents the adjoint solution for the case where the output functional is the volume integral of $S_w$ over the entire space-time domain $\Omega$.

\[ J_1 = \int_{\Omega} S_w(x, t) \, d\Omega \]  
(A.19)

\[ = \int_0^T \left( \int_0^L S_w(x, t) dx \right) \, dt \]  
(A.20)

Substitute the term inside brackets with the result shown in Eq. (A.12):

\[ J_1 = \int_0^T \left( \int_0^t \frac{u_T}{\phi} \, dt \right) \, dt \]  
(A.21)

\[ = \frac{1}{2} \frac{u_T}{\phi} T^2 \]  
(A.22)

Therefore this output $J_1$ is also a constant.

Derive the adjoint equation by considering infinitesimal perturbations to $S$, i.e. $\tilde{S}$:

\[ \tilde{J}_1 = \int_{\Omega} \psi \nabla \cdot \tilde{F} \, d\Omega \]  
(A.23)

\[ \int_{\Omega} \tilde{S}_w(x, t) \, d\Omega = \int_{\partial \Omega} \psi \tilde{F} \cdot n \, dA - \int_{\Omega} \nabla \psi \cdot \tilde{F} \, d\Omega \]  
(A.24)

Adjoint equation:

\[ \nabla \psi \cdot \tilde{F} + \tilde{S}_w = 0 \]  
(A.25)

\[ \phi \tilde{S}_w \frac{\partial \psi}{\partial t} + u_T f(\tilde{S}_w) \frac{\partial \psi}{\partial x} + \tilde{S}_w = 0 \]  
(A.26)

Adjoint BCs:

\[ \int_{\partial \Omega} \psi \tilde{F} \cdot n \, dA = 0 \]  
(A.27)

\[ - \int_0^L \psi(x, 0) \cdot \phi \tilde{S}_w(x, 0) \, dx + \int_0^L \psi(x, T) \cdot \phi \tilde{S}_w(x, T) \, dx \]  
(A.28)

\[- \int_0^T \psi(0, t) \cdot u_T f(\tilde{S}_w(0, t)) \, dt + \int_0^T \psi(L, t) \cdot u_T f(\tilde{S}_w(L, t)) \, dt = 0 \]
Noting that the perturbations $\tilde{S}_w = 0$ on the left boundary (i.e. $\tilde{S}(0, t) = 0$) and the bottom boundary (i.e. $\tilde{S}(x, 0) = 0$), and that $f(0) = 0$:

$$\int_0^L \psi(x, T) \cdot \phi \tilde{S}_w(x, T) \, dx - \int_0^T \psi(0, t) \cdot u_T f(0) \, dt + \int_0^T \psi(L, t) \cdot u_T f(0) \, dt = 0$$

$$\int_0^T \psi(x, T) \cdot \phi \tilde{S}_w(x, T) \, dx = 0$$

$$\implies \psi(x, T) = 0$$

(A.29)

The following linear solution for $\psi$ can be shown to satisfy the adjoint equation in Eq. (A.26) and the corresponding boundary condition in Eq. (A.29):

$$\psi(x, t) = \frac{1}{\phi}(T - t)$$

(A.30)

### A.3 Output estimates from a time-marching FV method

This section provides a proof of why the numerical result of $J_0$ is exact when computed using a 1D time-marching finite volume method. Let the FV estimate for $J_0$ be $\hat{J}_h$. Consider the Buckley-Leverett equation in the following form:

$$\frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x} \left( \frac{u_T}{\phi} f(S_w) \right) = 0$$

(A.31)

$$S_w(0, t) = S_L = 1 \quad t \in [0, T]$$

(A.32)

$$S_w(x, 0) = 0 \quad x \in [0, L]$$

The FV discretization of Eq. (A.31) is given in Eq. (A.33), which uses a first-order upwind numerical flux and a Backward-Euler time discretization.

$$\hat{S}_i^{n+1} = \hat{S}_i^n - \frac{\Delta t}{\Delta x} \left( \hat{F}_{i+\frac{1}{2}}^{n+1} - \hat{F}_{i-\frac{1}{2}}^{n+1} \right) \quad \text{for } i = 1, \ldots, N_x$$

(A.33)

$$\hat{S}_i^0 = 0 \quad \text{for } i = 1, \ldots, N_x$$

(A.34)

where $\hat{S}_i^n$ represents the average value of $S_w$ in cell $i$ at time level $n$ and $\Delta t$ is the size
of the timestep. The 1D domain is uniformly divided into \( N_x \) cells, each of length \( \Delta x \). \( \hat{F}_{i+\frac{1}{2}} = \hat{F}_{i+\frac{1}{2}}(\hat{S}_i, \hat{S}_{i+1}) \) is the numerical flux at the interface between cell \( i \) and cell \( (i + 1) \).

Sum Eq. (A.33) over all cells and multiply by \( \Delta x \) to obtain:

\[
\Delta x \sum_{i=1}^{N_x} \hat{S}_i^{n+1} = \Delta x \sum_{i=1}^{N_x} \hat{S}_i^n - \Delta t \sum_{i=1}^{N_x} \left( \hat{F}_{i+\frac{1}{2}}^{n+1} - \hat{F}_{i-\frac{1}{2}}^{n+1} \right) \tag{A.35}
\]

Noticing that all interior interface numerical fluxes cancel each other, and defining \( \hat{J}^n_h \) to be the integral of the FV solution \( \hat{S}_i \) over the domain at time level \( n \):

\[
\hat{J}^{n+1}_h = \hat{J}^n_h - \Delta t \left( \hat{F}_{N_x+\frac{1}{2}}^{n+1} - \hat{F}_{\frac{1}{2}}^{n+1} \right) \tag{A.36}
\]

Using the above boundary conditions and assuming that \( \hat{S}_1 \) and \( \hat{S}_{N_x} \) remain within the range 0 to 1, it is known that characteristics move in to the domain from the left boundary and move out of the domain at the right boundary. Furthermore, it is assumed that \( T \) is sufficiently small that \( \hat{S}_{N_x} = 0 \) for all \( t = [0, T] \). This is true for the test problem discussed in Chapter 5, since the saturation shock does not reach the right boundary in \( T = 30 \) days.

Hence the numerical fluxes at the boundaries can be evaluated as follows:

\[
\hat{F}_{\frac{1}{2}}^n = \frac{u_T}{\phi} f(S_L) = \frac{u_T}{\phi} f(1) = \frac{u_T}{\phi} \tag{A.37}
\]

\[
\hat{F}_{N_x+\frac{1}{2}}^n = \frac{u_T}{\phi} f(S_{N_x}) = \frac{u_T}{\phi} f(0) = 0 \tag{A.38}
\]

Eq. (A.36) can now be simplified to:

\[
\hat{J}^{n+1}_h = \hat{J}^n_h + \Delta t \frac{u_T}{\phi} \tag{A.39}
\]

Considering that \( \hat{J}^0_h = 0 \), the FV estimate for the integral of \( S_w \) at time level \( n \) is
given by:

\[ \dot{J}_h^n = n \Delta t \frac{uT}{\phi} \]  \hspace{1cm} (A.40)

Since \( J_{h0} \) is equivalent to \( \dot{J}_h^n \) at time level \( n = \frac{T}{\Delta t} \), the FV estimate for \( J_0 \) is equal to:

\[ J_{h0} = \left( \frac{T}{\Delta t} \right) \Delta t \frac{uT}{\phi} = \frac{uT}{\phi} T \]  \hspace{1cm} (A.41)

Comparing Eq. (A.41) with Eq. (A.12) shows that \( J_{h0} \) and \( J_0 \) are exactly equal to each other, independent of cell size \( \Delta x \) or timestep size \( \Delta t \).
Bibliography


135


