

RICE UNIVERSITY

Numerical Error Quantification of Agent-Based
Models as Applied to Oil Reservoir Simulation

by

Bryan Doyle

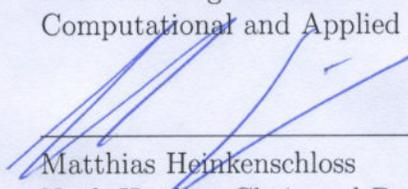
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Arts

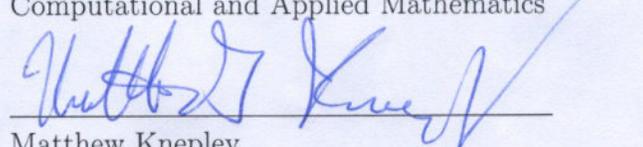
APPROVED, THESIS COMMITTEE:



Beatrice Riviere, Chair
Noah Harding Chair and Professor of
Computational and Applied Mathematics



Matthias Heinkenschloss
Noah Harding Chair and Professor of
Computational and Applied Mathematics



Matthew Knepley
Assistant Professor of Computational and
Applied Mathematics

Houston, Texas

December, 2017

ABSTRACT

Numerical Error Quantification of Agent-Based Models as Applied to Oil Reservoir Simulation

by

Bryan Doyle

Agent-based models (ABMs) provide a fast alternative to traditional oil reservoir models by applying localized inexpensive simulations, rather than solving a partial differential equation at every time-step. However, while there have been theoretical and numerical results obtained with ABMs in social science applications, the accuracy of ABMs has not been analyzed in the context of oil reservoir modeling. My project quantifies the accuracy of a specific ABM by comparing its results to a widely accepted reservoir model, based on Darcy's law. I show that while modeling single phase flow with a variety of reservoir scenarios, this ABM matches results given by the traditional simulator with less than 5.4% difference. I propose extensions of my work, including modeling two and three phase flow, and obtaining an accurate correlation between the ABM and traditional simulator parameters; such results would provide significant motivation in the extended use of ABMs in oil reservoir modeling.

Contents

Abstract	ii
List of Illustrations	v
List of Tables	x
1 Background and Literature Review	1
1.1 Introduction	1
1.2 Agent Based Models	2
1.2.1 Definition and History	2
1.2.2 Example: Application to Heat Equation	7
1.2.3 Application to Oil Reservoirs	9
1.3 Traditional Approaches to Oil Reservoir Modeling: Literature Review	14
2 Methods	16
2.1 Traditional Model: PDE	16
2.2 Traditional Model: Discretization	18
2.3 ReservoirGrail: Equalization Rate	25
2.4 Solution Conversion	26
3 Results	29
3.1 Test Setup	29
3.1.1 Data Analysis	29
3.1.2 Static Reservoir Variables	30
3.2 Computational Tests	32
3.2.1 Single and Double Producers	34

3.2.2	Varied Permeability	41
3.2.3	Impermeable Barrier with Single Producer	44
3.2.4	Double Impermeable Barrier	52
3.2.5	Small Permeability and EQ Rate Tests	66
3.2.6	Refined Mesh Tests	69
4	Future Work	78
4.1	Future Work	78
	Appendices	80
	A Tables for Reservoir Tests	81
	B Unit Descriptions and Conversions	109
	C ReservoirGrail: Additional Information	112
C.1	Importing Data	112
C.2	Defining Reservoir Maps	113
C.3	Defining Reservoir Parameters and Output Files	117
C.4	ReservoirGrail Flowchart	120
	Bibliography	123

Illustrations

1.1	Traditional model flowchart, utilizing linear systems (the bottleneck of such simulations).	4
1.2	Agent-based model flowchart utilizing the application of simple rules and agents, which allows such a simulation to execute faster than its traditional counterpart.	4
1.3	Ant colony model, $T = 0$ seconds.	7
1.4	Ant colony model, $T = 3$ seconds.	7
1.5	Ant colony model, $T = 13$ seconds.	7
1.6	Connectivity between the large, green central cell and its eight neighbors.	11
3.1	Domain of oil reservoir for single producer test. The black dot marks the location of the production oil well.	34
3.2	Domain of oil reservoir for double producer test. The black dots mark the locations of the two production wells.	35
3.3	Permeability value vs. ℓ_2 error, in percentage, for a single production well scenario. Note that the first data point is higher than the rest, and that the error monotonically increases after the ratio climbs above 40mD. See tables A.1 through A.9.	36
3.4	Permeability value vs. best match EQ rate, for a single production well scenario. Note that the best match EQ rate plateaus after the permeability climbs above 40mD. See tables A.1 through A.9.	37

3.5	Permeability value vs. ℓ_1 error, in percentage, for a single production well scenario. Same behavior as depicted in figure 3.3. See tables A.1 through A.9.	38
3.6	Permeability value vs. ℓ_∞ error, in percentage, for a single production well scenario. Same behavior as depicted in figure 3.3. See tables A.1 through A.9.	39
3.7	Permeability value vs. ℓ_2 error, in percentage, for a double production well scenario. Note that the first data point is higher than the rest, and that the error monotonically increases after the permeability climbs above 30mD. See tables A.10 through A.18. . . .	40
3.8	Permeability value vs. best match EQ rate, for a double production well scenario. Note that the best match EQ rate plateaus after the permeability climbs above 30mD. See tables A.10 through A.18. . . .	41
3.9	Permeability value vs. ℓ_1 error, in percentage, for a double production well scenario. Same behavior as depicted in figure 3.7. See tables A.10 through A.18.	42
3.10	Permeability value vs. ℓ_∞ error, in percentage, for a double production well scenario. Same behavior as depicted in figure 3.7. See tables A.10 through A.18.	43
3.11	Cell by cell percent error of the single producer test.	44
3.12	Cell by cell barrels of oil error of the single producer test.	45
3.13	Cell by cell percent error of the double producer test.	46
3.14	Cell by cell barrels of oil error of the double producer test.	47
3.15	Domain of 40 acre test with a single well. An area of 40 acres corresponds to a circle with radius 750 feet.	48
3.16	Domain of 40 acre test with a single well. An area of 40 acres corresponds to a circle with radius 750 feet.	49

3.17	The domain for the varied permeability test. The green area has a permeability of 200mD, while the white area has a permeability of 10mD.	52
3.18	Cell by cell percent error of varied permeability test.	53
3.19	Cell by cell barrels of oil error of varied permeability test.	54
3.20	Connectivity between the large, green central cell and its neighbors, with a red barrier severing some of the original connections. In this example, ReservoirGrail models the barrier as a one dimensional object.	54
3.21	Connectivity between the two large green cells and their neighbors, with a barrier severing some of their connections. In this example, ReservoirGrail models the barrier as a two dimensional object, as the red cells in the middle are essentially removed from the simulation. .	55
3.22	The domain for the single barrier test. Red cells represent the location of the barrier.	56
3.23	Cell by cell percent error of the single barrier, 30 mD test.	57
3.24	Cell by cell barrels of oil error of the single barrier, 30 mD test. . . .	58
3.25	Cell by cell percent error of the single barrier, 50 mD test.	59
3.26	Cell by cell barrels of oil error of the single barrier, 50 mD test. . . .	60
3.27	Cell by cell percent error of the single barrier, 100 mD test.	61
3.28	Cell by cell barrels of oil error of the single barrier, 100 mD test. . . .	62
3.29	The domain for the double barrier, single producer test. Red cells represent the location of the barrier.	63
3.30	Cell by cell percent error of the double barrier, single producer test. .	64
3.31	Cell by cell barrels of oil error of the double barrier, single producer test.	65
3.32	The domain for the double barrier, double producer test with varied permeability and a barrier to the left of both production wells.	66
3.33	Cell by cell percent error of the double barrier, double producer, varied permeability test, with a left barrier.	67

3.34	Cell by cell barrels of oil error of the double barrier, double producer, varied permeability test, with a left barrier.	68
3.35	The domain for the double barrier, double producer test with varied permeability and a barrier in between both production wells.	69
3.36	Cell by cell percent error of the double barrier, double producer, varied permeability test, with a central barrier.	70
3.37	Cell by cell barrels of oil error of the double barrier, double producer, varied permeability test, with a central barrier.	71
3.38	EQ rate vs. ℓ_∞ error of the 10mD permeability test.	72
3.39	Comparison between ℓ_∞ error when a coarse EQ rate step is used vs. a finer step.	73
3.40	EQ rate vs. ℓ_∞ error of the 8mD permeability test.	74
3.41	EQ rate vs. ℓ_∞ error of the 12mD permeability test.	74
3.42	Domain for the refined mesh test, with constant 30mD permeability.	75
3.43	Relationship between EQ rate and $\%l_2$ difference between the simulators, for the coarse and fine mesh; see table A.29 for more details.	75
3.44	Domain for the refined mesh test, with constant 30mD permeability and a single barrier.	76
3.45	Relationship between EQ rate and $\%l_2$ difference between the simulators, for the coarse and fine mesh with a single barrier; see table A.30 for more details.	77
B.1	Oil compressibility (y-axis), as a function of pressure (x-axis) [16].	110
C.1	Dialogue box, prompting the user to import data for their project. It is easier to import this data now, rather than after the project has been created.	113

C.2	Example of importing header data, which defines the locations and APIs of all production and injection wells in the reservoir.	114
C.3	The header data used, for all single producer tests.	114
C.4	Example of importing production well data.	115
C.5	The production data used, for all single producer tests.	116
C.6	ReservoirGrail's home screen.	116
C.7	Screen where local reservoir features are specified, such as fault lines and permeability changes.	117
C.8	The fault line data used, for the single barrier tests.	118
C.9	Screen where global reservoir parameters are specified, such as porosity and initial oil and gas saturation.	119
C.10	Options to determine what outputs ReservoirGrail generates for the simulation.	119
C.11	A flowchart of the well phase of ReservoirGrail's algorithm, which is used to send out agents and record their visits across the domain. . .	121
C.12	A flowchart of the production phase of ReservoirGrail's algorithm, which is used to remove the appropriate amount of hydrocarbons from each cell and equalize the remaining oil, gas, and water as necessary. .	122

Tables

3.1	Variables and their values in the PDE model.	31
3.2	Variables and their values in the agent-based model.	31
3.3	Variables and their values that are common to both models.	33
3.4	The difference between Darcy’s law simulator and ReservoirGrail, for a single well. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	50
3.5	The difference between Darcy’s law simulator and ReservoirGrail, for two wells. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	51
3.6	Percent errors for the varied permeability test. Note that while the values are slightly higher than in Section 3.2.1, they are still relatively low.	51
3.7	The difference between the traditional and ReservoirGrail simulators, for the variable permeability test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	52
3.8	Percent errors for the single barrier, 30 mD test. Note that while the values are slightly higher than in Section 3.2.1, they are still relatively low.	53

3.9	The difference between the traditional and ReservoirGrail simulators, for the single barrier, 30 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	55
3.10	Percent errors for the single barrier, 50 mD test.	56
3.11	The difference between the traditional and ReservoirGrail simulators, for the single barrier, 50 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	57
3.12	Percent errors for the single barrier, 100 mD test.	58
3.13	The difference between the traditional and ReservoirGrail simulators, for the single barrier, 100 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	59
3.14	Percent errors for the double barrier, single producer test.	60
3.15	The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, single producer test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	61
3.16	Percent errors for the double barrier, double producer, varied permeability test, with a left barrier.	62
3.17	The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, double producer, varied permeability test, with a left barrier. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	63
3.18	Percent errors for the double barrier, double producer, varied permeability test, with a central barrier.	64

3.19	The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, double producer, varied permeability test, with a central barrier. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.	65
3.20	Percent errors for the constant permeability of 30mD test, with both a coarse and fine mesh.	72
3.21	Percent errors for the single fault test, with both a coarse and fine mesh.	73
A.1	Error comparisons between different EQ rates, for the single producer scenario with permeability 10mD.	81
A.2	Error comparisons between different EQ rates, for the single producer scenario with permeability 20mD.	82
A.3	Error comparisons between different EQ rates, for the single producer scenario with permeability 30mD. *: Three decimal places were given here in order to show which EQ rate minimized the given norm; with only two decimal places, this determination would be non-unique. . .	83
A.4	Error comparisons between different EQ rates, for the single producer scenario with permeability 40mD.	84
A.5	Error comparisons between different EQ rates, for the single producer scenario with permeability 50mD.	85
A.6	Error comparisons between different EQ rates, for the single producer scenario with permeability 100mD.	86
A.7	Error comparisons between different EQ rates, for the single producer scenario with permeability 200mD.	87
A.8	Error comparisons between different EQ rates, for the single producer scenario with permeability 500mD.	88

A.9 Error comparisons between different EQ rates, for the single producer scenario with permeability 1000mD.	89
A.10 Error comparisons between different EQ rates, for the double producer scenario with permeability 10mD.	90
A.11 Error comparisons between different EQ rates, for the double producer scenario with permeability 20mD.	91
A.12 Error comparisons between different EQ rates, for the double producer scenario with permeability 30mD. *: Three decimal places were given here in order to show which EQ rate minimized the given norm; with only two decimal places, this determination would be non-unique.	92
A.13 Error comparisons between different EQ rates, for the double producer scenario with permeability 40mD.	93
A.14 Error comparisons between different EQ rates, for the double producer scenario with permeability 50mD.	94
A.15 Error comparisons between different EQ rates, for the double producer scenario with permeability 100mD.	95
A.16 Error comparisons between different EQ rates, for the double producer scenario with permeability 200mD.	96
A.17 Error comparisons between different EQ rates, for the double producer scenario with permeability 500mD.	97
A.18 Error comparisons between different EQ rates, for the double producer scenario with permeability 1000mD.	98
A.19 Error comparisons between different EQ rates, for the varied permeability test.	99
A.20 Error comparisons between different EQ rates, for the single barrier, 30mD test.	100

A.21 Error comparisons between different EQ rates, for the single barrier, 50mD test.	101
A.22 Error comparisons between different EQ rates, for the single barrier, 100mD test.	102
A.23 Error comparisons between different EQ rates, for the double barrier, single producer, 30mD test.	103
A.24 Error comparisons between different EQ rates, for the double barrier, double producer, variable permeability with a left barrier test.	104
A.25 Error comparisons between different EQ rates, for the double barrier, double producer, variable permeability with a central barrier test.	105
A.26 Error comparisons between different EQ rates, for the single barrier and 8mD permeability test.	105
A.27 Error comparisons between different EQ rates, for the single barrier and 10mD permeability test.	106
A.28 Error comparisons between different EQ rates, for the single barrier and 12mD permeability test.	106
A.29 Error comparisons between different EQ rates, for the refined traditional model mesh with constant 30mD permeability.	107
A.30 Error comparisons between different EQ rates, for the refined traditional model mesh with constant 30mD permeability and a single barrier.	108

Chapter 1

Background and Literature Review

1.1 Introduction

This thesis quantifies the accuracy of an agent-based model (ABM) in its application to oil reservoir modeling by comparing it to a traditional and widely accepted reservoir model. Agent-based models provide a fast alternative to traditional oil reservoir simulators, but precise bounds on their accuracy have proven difficult to obtain for all but a small subset of ABMs. This thesis takes an alternative route in quantifying the accuracy of such models by focusing on the rigorous computational testing of a specific ABM called ReservoirGrail. Utilizing standard error computation techniques, I determine to what extent ReservoirGrail captures the expected global and local behavior of a given reservoir over time, by comparing ReservoirGrail's results to those of a traditional oil reservoir model whose accuracy is known.

The traditional model I choose to implement solves the compressible single-phase Darcy equation using a discontinuous Galerkin discretization in space and modified backward Euler as a time-step method. Through this testing, I show that for single phase flow in both simple and more complex reservoir setups, the error between ReservoirGrail and the traditional model is low, which suggests that ReservoirGrail produces results that are exceptionally close to those of a trusted oil reservoir simulator. Furthermore, I determine a correlation between a key artificial parameter within ReservoirGrail and physical parameters that are required in the traditional model.

Agent-based models have previously been widely used in the social sciences (Conte & Paolucci [4]), and while they have recently been used to model more scientific problems (Azimi, Jamali, & Mofrad [5]), they had previously not been applied and tested in the specific scenario of oil reservoir modeling.

1.2 Agent Based Models

Agent-based models are generic in their definition and wide-spread in their application. This section is intended to serve as an introduction to ABMs, as both a general tool as well as to their specific application to partial differential equations (PDEs) and oil reservoir modeling. I first introduce the general properties of agent-based models, as well as their past uses in modeling complex phenomena, in Section 1.2.1. Next, I restrict my attention to the application of an agent-based model in the context of the heat equation in Section 1.2.2, in order to depict how such a model is built in an equation-based system. Finally, I explain how ReservoirGrail is used in its application to oil reservoirs in Section 1.2.3, and describe how it works in the best detail I can, while respecting the proprietary nature of the model.

1.2.1 Definition and History

Agent-based modeling is a method of simulating the dynamics of complex systems, such as the interaction between the immune system and foreign bacteria in a biological model, or the interaction between consumers and their purchasing habits in an economics model. Low level entities in the simulation, called “agents”, represent granular, autonomous beings in the model. In a medical application, these agents could be cells in the system, which can react on their own in different ways to the presence of foreign objects. In an economic setting, the agents could be consumers,

who act and purchase items independently in accordance with their programmed behavior. There is no universal agreement in literature on the exact definition of an agent in an ABM, other than this property of autonomy (Macal, [13]). The behavior of agents is defined by a set of logical rules that all agents in the system follow; the specific rules of the model depend on the setting, but are generally obtained through condensing the data obtained from prior experiments or known governing equations into logical rules (Folick, [12]). Executing an agent-based model consists of having agents repeatedly applying their behaviors and interactions, with other agents and with their environment. This process generally takes place over a time interval, as the agents interact and evolve over time, although this does not necessarily have to be the case (Macal, [13]).

Agent-based models are generally faster than traditional models, due to their replacement of the linear systems with the simple logical rules. Furthermore, if the problem grows beyond the capability of a single processor, previous work has shown that it is possible to parallelize an agent-based algorithm to run across multiple processors (Kurve, Kotobi, & Kesidis [9]). The flowcharts in figures 1.1 and 1.2 depict the basic flow of processes between the two models over a single time-step, with the main difference between the two being the linear systems in the ABM, and the linear system in the traditional model. This speedup potential is what has driven the advancement and continued usage of agent-based models in a variety of different fields.

Historically, agent-based models arose in the early 1990's in order to model social and behavioral interactions between a large number of individuals. The first major program to utilize ABMs at its core was called Sugarscape, and was used to model migrations of animals, pollution, and disease propagation to name a few (Epstein & Axtell, [7]). While ABMs were not immediately accepted in the social science com-

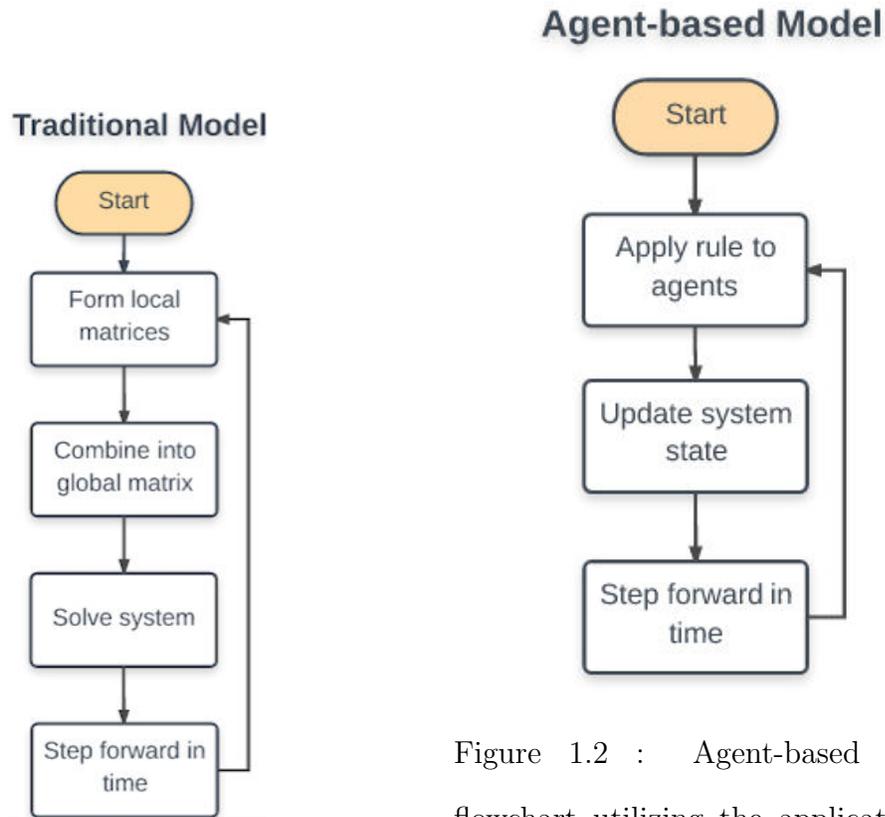


Figure 1.1 : Traditional model flowchart, utilizing linear systems (the bottleneck of such simulations).

Figure 1.2 : Agent-based model flowchart utilizing the application of simple rules and agents, which allows such a simulation to execute faster than its traditional counterpart.

community, such models started to gain reputation in the early to mid 2000s, and more conference talks and journal articles started utilizing them in one manner or another. Shortly thereafter in 2010, the concept of agent-based models in their application to social sciences was reenvisioned to include more complex systems, and as a result, a more elaborate class of models was introduced called “Computational Social Science”. These new models have the same basic concepts at their core, in that they model numerous local entities to predict global behavior of systems, but can take into

account many additional inputs that influence social science directly, such as the cultural background or economic status of the agents (e.g., people) in the system (North, [4]). For this reason, the terminology of “agent-based models” in social sciences has been predominantly replaced with the broader term “computational social science”.

Within the last 10 years, agent-based models have been applied with more frequency to a larger number of the “hard” sciences. In 2007 an ABM was used in a biological setting, reproducing clinical data focusing on diabetic foot ulcers, and was further utilized for testing hypothesis regarding the cause of such ulcers as well as the effects of common therapeutical techniques (Mi, [11]). Furthermore, an agent-based model called the “Basic Immune Simulator” was created in 2007 in order to simulate and study the interactions between various types of cells in the immune system, in their response to a pathogen (Folick, [12]). This ABM was shown to effectively capture the normal and pathological behavior in different viral infection scenarios.

ABMs have also been utilized to model systems that are stochastic in nature. In 2008, a study was performed comparing an agent-based model and a deterministic (PDE based) model in the context of disease propagation and it was found that while both models yielded realistic results, the ABM was able to give more probabilistic information (expected outcome, variance in results, and result distribution) due to the stochastic nature of the underlying agents, whereas the deterministic model only gave a single outcome based on its input parameters (Rahmandad & Sterman, [14]). Eight years later in 2016, a hybrid model derivation was given in the context of vocal fold inflammation and healing, which utilized a mathematical approach to solve for the chemical diffusion in the system, but an agent-based approach to model the cellular interactions (Seekhao, Shung, Jaja, Mongeau, & Li-Jessen [15]). This paper also went further to give a parallelizable and generic implementation of their algorithm,

providing a tractable starting point for future research in this direction.

The acknowledged downside of agent-based models is their lack of substantial theoretical backing. A report in 2013 provided theoretical findings on the convergence rates of ABMs in their application to a fairly broad class of complex, adaptive systems (North, [3]), but other than this, there are few globally applicable results pertaining to the convergence and accuracy of ABMs. In stark contrast, the methods used to solve equation based models (such as the finite element method) have deep roots in mathematical theory, and as such, have traditionally been the preferred method to solve equation based systems (I go into more detail about traditional modeling methods in Section 1.3). Despite this, a 2011 paper described an ABM applied to a diffusion-reaction system in its modeling to cytoskeletal diffusion, and showed how the governing PDE can be used to inform the agents' logical rules (Azimi, Jamali, & Mofrad [5]). In this scenario, the agents were chosen to represent the cells in the system, and moved around in the domain with probability influenced by parameters present in the diffusion-reaction PDE, such as the diffusivity coefficient and density of cells in the given area. For all their past uses in the scientific community, however, the effectiveness and overall accuracy of applying agent-based models to general equation based systems has not been explored, and due to the lack of universal theoretical results of ABMs, their error cannot be quantified a-priori. For this reason, the application of ABMs to equation based systems is an underexplored area of research.

Figures 1.3 through 1.5 ([17]) depict an example of how an agent-based model can evolve through time, and were obtained from an algorithm that models the movement of ants in two different colonies, using agent-based modeling techniques. In this simulation, the agents are modeled as ants, and the rules that they follow are obtained from observed, real-world behavior of ants; that is, the ants are programmed to search

for food, return it to their colony, tell other ants where the food is located, and defend themselves from ants of another colony. The blue and green dots in figures 1.3 through 1.5 represent ants of two different colonies, the brown dots represent food sources, the light blue and dark green swaths of the domain represent pheromones that the ants emit when they have found food, and the red areas represent areas where ants of two different colonies are fighting one another.



Figure 1.3 : Ant colony model, $T = 0$ seconds.

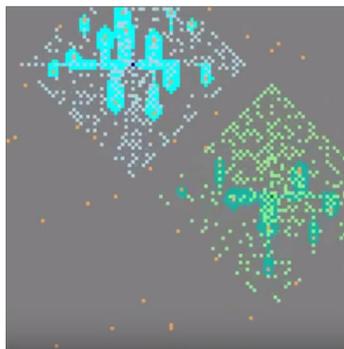


Figure 1.4 : Ant colony model, $T = 3$ seconds.

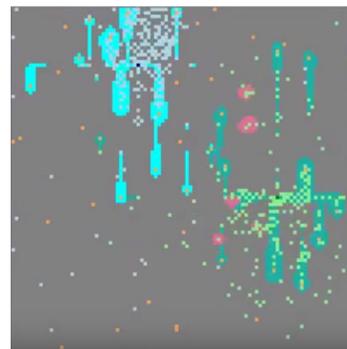


Figure 1.5 : Ant colony model, $T = 13$ seconds.

1.2.2 Example: Application to Heat Equation

Natural questions arise when applying an ABM to an equation based system with no obvious low-level entities, such as the heat equation: what are the agents in such a formulation? What rules do they follow? How do agent movements correlate to the rate at which the heat diffuses across the domain? This section will show one possible way that an ABM can be built to model the heat diffusion in a domain, using characteristics found in the governing PDE.

Consider the heat equation, on a time interval of $[0, T]$ and a domain Ω :

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (k \nabla u) &= f \text{ in } \Omega \times [0, T], \\ k \nabla u \cdot \mathbf{n} &= 0 \text{ in } \partial\Omega \times [0, T], \\ u(\mathbf{x}, 0) &= u_{\text{init}}, \end{aligned} \tag{1.1}$$

where k is the thermal conductivity of the domain, f is the source term, and u_{init} is the initial condition. For simplicity, suppose f is a point source in Ω . In order to use an ABM in this context to model the diffusion of heat throughout a domain, the agents and rules that they follow must be clearly defined; in this case, agents and their rules are used to inform the diffusion of heat throughout the domain as time progresses, and do not represent any physical entity in the system. This is in stark contrast to typical agent-based modeling, where the agents and their rules correspond to physical and observable characteristics of the system being modeled.

In order to use agents to accurately inform how the heat moves around in the domain, the physical characteristics of the system (namely, k and f) must be taken into account. One approach to best utilize these variables is to consider agents that start at the point source f , and move around the domain stochastically, influenced by the local conductivity values; that is, as an agent moves throughout the domain, the probability that it will move in a given direction will be lower if the thermal conductivity in that direction is lower (relative to other possible directions). Once an agent has taken a specified number of “steps”, it is retired, another agent is sent out, and the process is repeated. Repeating this process for a large number of agents will generate a map of agent visits throughout the domain: areas of the domain with a higher number of visits correspond to areas that are both close to the point source, as well as areas that have a higher thermal conductivity. Therefore, areas with a higher number of visits will correlate to areas that the heat will naturally move to. In this

way, the agents are used to inform the heat transfer throughout the domain, without correlating to any kind of physical quantity.

Due to the absence of the practical and theoretical application of ABMs to equation governed systems, there has been little academic or industrial consideration in the specific application of agent-based models to oil reservoir systems. It is for this reason that in Section 1.2.3, I turn my attention to defining an ABM of this form; that is, an ABM that models a PDE based oil reservoir, and whose agents and rules do not correlate to any kind of physical quantity.

1.2.3 Application to Oil Reservoirs

The application of ABMs to oil reservoirs is similar in its application to the heat equation in Section 1.2.2, as the PDE that governs the pressure in reservoirs is also parabolic in nature and behaves similarly.

The specific ABM I use to model oil reservoirs is called ReservoirGrail, developed by GrailQuest ([19]). ReservoirGrail initially requires two maps of the reservoir being modeled: the flow-path map, which depicts the distribution of flow characteristics throughout the reservoir*, and a pay thickness map, depicting the thickness of the reservoir that is able to produce oil and gas at each node. However, these maps do not have to be exact: ReservoirGrail has the functionality to accept past production and injection data from the wells also, and uses this to modify the given parameters so that the simulation matches up better with the past data. In this manner, ReservoirGrail is suited well as a mature field simulator where many years of production data are available.

*This flow path map can be any map the user has that contains information concerning, flow characteristics, such as a permeability map, porosity map, or seismic map.

Once ReservoirGrail has decided on a suitable set of parameters for the simulation, an agent-based algorithm ([18]) is applied to simulate the future movement of subsurface hydrocarbons in the reservoir. The exact set of rules that are applied to the agents is proprietary knowledge to GrailQuest; however, the agents in this model behave in such a way to move fluid from high pressure to low pressure as quickly as possible, while taking into account the global structure of the reservoir, such as impermeable barriers and relative permeability changes across the domain. A sketch of ReservoirGrail's algorithm is discussed below. Although ReservoirGrail can be applied to real, 3D reservoirs, the algorithm it uses models the hydrocarbon movement in a 2D domain, then extends its results to 3D by utilizing the user provided thickness map.

ReservoirGrail has two main phases in its simulation process, for every time-step: the production phase, which determines the amount of hydrocarbons removed from each cell due to the production rate of the wells, and the equalization phase, which models the movement of the hydrocarbons after the production phase is complete.

The production phase utilizes agent-based modeling to determine the volumetric depletion of every cell, considering a single well at a time. First, a number of agents (on the order of 10^5) are sent out starting from the first well in the reservoir, and move stochastically around the reservoir. The actual number of agents that are sent out can be a set value, or can depend on the relative production of the well compared to all the other wells in the reservoir. Every time the agent moves to a new cell, that new cell must be one of its neighbors. For an interior cell with no barriers present, there are eight potential neighbors for the agent to move to; see figure 1.6.

The probability that an agent moves to one of its neighbors can be calculated in a variety of ways. One such method utilizes the magnitude of the permeability

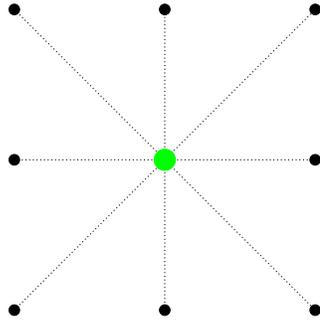


Figure 1.6 : Connectivity between the large, green central cell and its eight neighbors.

of the agent's neighbors, relative to each other: let A_E be the event that the agent moves to cell E , let K_E be the permeability of cell E , and let \mathcal{N} be the set of all of the agent's neighbors it is allowed to move to. Then, the probability of event A_E occurring, denoted $P(A_E)$, is calculated as

$$P(A_E) = \frac{K_E}{K_T}, \text{ where } K_T = \sum_{N \in \mathcal{N}} K_N.$$

Agents stop moving after a pre-determined number of steps; however, in some scenarios, additional stopping criterion can be imposed, such as when an agent reaches the boundary of the domain, or when the agent becomes surrounded by cells it has already visited.

After all agents have reached an appropriate stopping criterion, the depletion volume for each cell is calculated based on both the well production, and the fraction of the total number of times agents visited the cell to the total number of visits to all cells in the domain. For a given well W , let q_W denote the well production, let B_E denote the total number of times that an agent has visited cell E in the current time-step, and let \mathbb{E} be the set of all cells in the domain. Then, the depletion volume

of a cell E , denoted $(\Delta V)_E$, is calculated as

$$(\Delta V)_E = \frac{B_E}{B_T} q_W, \text{ where } B_T = \sum_{E' \in \mathbb{E}} B_{E'}.$$

After the depletion volume is calculated and removed for all cells in the domain, ReservoirGrail moves on to the equalization phase. First, a ratio called the “fill” of each cell is calculated, which represents how full the cell is of movable hydrocarbons, in percentage form. Specifically, let F_E denote the fill of cell E , let O_E denote the percentage of void volume (i.e., the percentage volume not taken up by hydrocarbons) in cell E , and let M_E denote the percentage of max void volume in cell E (i.e., the percentage of removable fluid from the cell, not including residual oil). Then, the fill of cell E is calculated as

$$F_E = 1 - \frac{O_E}{M_E}.$$

Once this ratio is calculated for all cells, ReservoirGrail goes through the domain cell by cell to determine how the remaining hydrocarbons equalize across the domain. To do so, a cell E is fixed and the set of all neighbors of E , denoted \mathcal{N}_E , is generated. Note that by definition, $E \in \mathcal{N}_E$. Using E and its neighbors, the average fill is calculated by a weighted combination of the cell’s fill values. Let \bar{F}_E represent the average fill of cell E and its neighbors; then, the average fill is given as

$$\bar{F}_E = \sum_{E' \in \mathcal{N}_E} F_{E'} M_{E'} w_{E'}, \text{ where } w_{E'} = \begin{cases} \frac{1}{\sqrt{2}} & E' \text{ is diagonally adjacent to } E \\ 1 & \text{else} \end{cases}.$$

This definition of the weights $w_{E'}$ is to ensure that cells that are further away from E (that is, diagonally adjacent) contribute less to the neighborhood average.

After calculating the average fill of cells in a neighborhood of E , ReservoirGrail determines how much fluid should be moved in or out of all cells in \mathcal{N}_E by comparing

the fill and average fill values, on a cell by cell basis. For a cell $E' \in \mathcal{N}_E$, if $F_{E'} < \bar{F}_E$, it means that the cell E' should receive some fluid from other cells in \mathcal{N}_E ; analogously, if $F_{E'} > \bar{F}_E$, it means that the cell E' should lose fluid. The amount by which E' gains or loses fluid is based on the equalization (EQ) rate, which is a parameter defined by the user prior to executing the simulation. In comparison to traditional reservoir simulators, this parameter behaves analogously to the ratio between permeability and viscosity, and its value greatly influences how the modeled reservoir evolves over time; see Section 2.3 for more information about the EQ rate.

Let $L_{E'}$ be the amount of fluid in cell $E' \in \mathcal{N}_E$, denote the change in fluid of cell E' by $(\Delta L)_{E'}$, and EQ denote the equalization rate; then, the change in fluid in cell E' is calculated as

$$(\Delta L)_{E'} = EQ (\bar{F}_E - F_{E'}) M_{E'} w_{E'}.$$

Therefore, an equalization rate of 1 corresponds to a reservoir which allows a larger amount of fluid movement between cells, whereas an equalization rate of 0 corresponds to a reservoir with no fluid movement in the equalization phase. The change in fluid is calculated and applied to every cell in \mathcal{N}_E , and this process is repeated until all cells in the domain have been considered. From there, the next well in the reservoir is fixed and the entire process is repeated. When all wells have been considered, the time step is incremented by 1, and the process starts over from the production phase.

Because there is no published literature on the application of agent-based model to oil reservoir modeling, certain guarantees on the accuracy of such a method must be investigated. Since theoretical error guarantees have proven difficult to come by in the past, the goal of my thesis is to computationally quantify the accuracy of ReservoirGrail through repeated testing and comparison with a traditional oil reservoir model, which have many theoretical guarantees concerning their accuracy.

Because many such PDE based models exist, I narrow my focus to ones that are commonly used in oil reservoir modeling, in order to have an unbiased comparison of the final results.

1.3 Traditional Approaches to Oil Reservoir Modeling: Literature Review

Traditionally, oil reservoirs are modeled by numerically solving a PDE which governs the mass conservation of sub-surface components. Previously built traditional models are analyzed in this section, and the model I use in my thesis work will be described in more detail in Section 2.1. To determine the specific simulator to use, there are three main questions to answer: what PDE will be modeled, how will the solution be approximated in space, and how will the solution be stepped forward in time.

Multi-phase fluid flow and transport equations can be used to solve for the saturation of each individual phase of the fluid throughout time (Jenny, Lee, & Tchelepi [8]). Alternatively, problems dealing with fluid flow may be cast in the form of a scalar conservation law to determine velocity of the fluid within the domain (Zhang & Shu [2]). Finally, a more simple single-phase model may be viewed in terms of the pressure in the domain via the single-phase Darcy equations.

Discretization in the spatial dimension is typically done utilizing one of two methods. The Finite Volume Method (FVM) is one such method, and past research has shown that for multi-phase fluid flow and transport equations, such a discretization is computationally scalable (Jenny, Lee, & Tchelepi [8]). Another popular discretization technique is the discontinuous Galerkin (DG) method; prior work has shown that for scalar conservation laws, this method provides optimal convergence for smooth flux

terms (Zhang & Shu, [2]). Furthermore, for general non-linear parabolic equations (such as the single-phase Darcy equation), DG provides optimal convergence in the H_1 norm (Riviere & Wheeler, [6]).

Finally, several time-stepping methods have been considered in prior work. The third order Runge-Kutta method (RK3) is one such method, and gives both stability (under certain time constraints on the time-step) and 3rd order global convergence (Zhang & Shu, [2]). Another choice is the Backward-Euler (BE) method, which is unconditionally stable regardless of the time-step, and converges linearly in time (Riviere & Wheeler, [6]).

Overall, traditional oil reservoir models are built by determining a PDE to govern the system, a discretization method in space, and a way to step forward in time. However, in spite of these distinctions, all such models fundamentally rely on constructing and solving large linear systems at each time step. Even though there are many areas of research dedicated to speeding up the process of forming and solving of these linear systems, it remains today the bottleneck of such methods. It is this area from which ABMs gain the most ground over traditional models.

Chapter 2

Methods

This chapter goes into details concerning the traditional model I used, as well as how I compared parameters between ReservoirGrail and the traditional model. I start in Section 2.1 by describing the partial differential equation that the traditional model solves. Then, in Section 2.2, I defend my choice of spatial and temporal discretization for the model, referring heavily back to Section 1.3. Next, I describe in Section 2.3 how a correlation between ReservoirGrail’s artificial parameter and the traditional model’s parameters was obtained. Finally, in Section 2.4, I give a mathematical formulation for the conversion from pressure to mass in the traditional model, and from hydrocarbon pore volume to mass in ReservoirGrail, in order to compare results between the two.

2.1 Traditional Model: PDE

As mentioned in Section 1.3, there are three questions which govern the decision as to which traditional simulator to use: the governing PDE, the approximation method in space, and the time-stepping method. Because it is the goal of this thesis to provide preliminary results on the accuracy of an ABM, I model a system with lower complexity and which requires fewer computational resources, at the drawback of sacrificing overall robustness. Therefore, I utilize the single phase Darcy equation in the traditional simulator, which is both relatively simple to implement and has a low

computational cost:

$$\begin{aligned} \frac{\partial}{\partial t}(\phi\rho) - \nabla \cdot \left(\rho \frac{K}{\mu} \nabla p \right) &= \rho f \text{ in } \Omega \times (0, T), \\ \rho \frac{K}{\mu} \nabla p \cdot \mathbf{n} &= 0 \text{ in } \partial\Omega \times (0, T), \\ p(x, 0) &= p_{\text{init}}. \end{aligned} \tag{2.1}$$

In this equation, the variable p represents the pressure, and is taken to be the unknown in the system. The symbol Ω is used to denote the domain on which the equation is solved over, the symbol $\partial\Omega$ is used to denote the boundary of the domain, and T is used to denote the final time the model is solved up to. The variable ϕ represents the porosity of the reservoir, and for a fixed area of the reservoir, is defined as the ratio between the volume that hydrocarbons can occupy (i.e., volume that is not occupied by rock), and the total volume:

$$\phi = \frac{\text{void volume}}{\text{total volume}}$$

The porosity is a unitless number, and always takes values between 0 and 1. The variable ρ represents the density of the fluid being modeled. In this formulation, an ideal fluid density approximation is used for the density:

$$\rho = \rho_{\text{ref}}(1 + cp).$$

With this substitution, the value ρ_{ref} represents the reference density of the fluid; that is, the density of the fluid at standard temperature and pressure (STP). The variable c represents the compressibility of the fluid, and its introduction is why this formulation is a compressible single phase flow model, as opposed to an incompressible model. In the first line of equation (2.1), the variable K represents the permeability of the rock in the reservoir, and μ represents the viscosity of the fluid. The smaller the permeability or the higher the viscosity values are, the slower the fluid will flow

through the reservoir. Note that while permeability and porosity may appear to be similar variables, they are inherently different: porosity is a measure of free space available and is directly related to the reservoir texture, whereas permeability is a measure of how much fluid movement the rock in the reservoir allows for, and is directly related to the type of rock in the reservoir. Finally, the variable f represents the source term in the model, and models the actions of the wells in the reservoir (if any). A negative source term correlates to a production well (i.e., a well that is removing hydrocarbons from the reservoir), and a positive source term correlates to an injection well (i.e., a well that is injecting fluid into the reservoir).

For the purposes of this model, porosity ϕ and compressibility c will be assumed to be constant with respect to time. Applying this assumption and the ideal fluid density approximation, and dividing out the constant ρ_{ref} in equation (2.1), the final PDE to be solved is obtained:

$$\begin{aligned} \phi c \frac{\partial p}{\partial t} - \nabla \cdot \left((1 + cp) \frac{K}{\mu} \nabla p \right) &= (1 + cp) f \text{ in } \Omega \times (0, T), \\ (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} &= 0 \text{ in } \partial\Omega \times (0, T), \\ p(x, 0) &= p_{\text{init}}. \end{aligned} \tag{2.2}$$

2.2 Traditional Model: Discretization

After deciding upon the single-phase pressure model, I chose to use a discontinuous Galerkin method in space. Across the literature I found, the two most popular spatial discretization techniques were the finite element method (FEM), and the finite volume method (FVM). In some circumstances, FVMs and FEMs have been found to perform similarly for certain PDEs (Fallah, Bailey, Cross, & Taylor, [10]). However, in the specific application of reservoir modeling, it has been shown that FVMs yield results that are more diffusive in nature than DG methods, especially around the propagation

front (Li, [21]), due to the artificial diffusion introduced by the formulation of said FVMs. Due to the enhanced simulation results obtained when utilizing a DG method over a FVM, I chose to discretize with DG in space.

Finally, the time-stepping technique I chose to use was Backward Euler. There are several time-stepping techniques utilized in the literature, but many papers closely related to oil reservoir simulation tend to use either a 3rd order Runge-Kutta method (RK3) or a Backward Euler method. As discussed in Section 1.3, RK3 exhibits 3rd order convergence, while BE only exhibits 1st order convergence. However, BE is both unconditionally stable (unlike RK3) and is significantly easier to implement and debug. Again, because the goal of this model is to stay as simple as possible while still maintaining functionality, I chose to utilize BE for time-stepping.

Equation (2.2) will now be discretized with these two methods. First, I derive the weak form of equation (2.2). Next, I rewrite one of the summations as a sum over edges, instead as a sum over elements, using the divergence theorem. Afterwards, I add the penalty terms, and finally, I discretize with the backward Euler method and simplify the equation.

Let \mathcal{T}_h be a shape-regular mesh of Ω , and let Γ_h be the set of all edges (interior and on the boundary) of the partition. Define the finite dimensional space $\mathcal{D}_k(\mathcal{T}_h)$, known as the broken polynomial space of degree k , to be:

$$\mathcal{D}_k(\mathcal{T}_h) = \{v \in L_2(\Omega) \mid \forall E \in \mathcal{T}_h, v|_E \in \mathbb{P}_k(E)\},$$

where $\mathbb{P}_k(E)$ is the k degree polynomial space over the element E .

Fix an element $E \in \mathcal{T}_h$, multiply equation (2.2) by $v \in \mathcal{D}_k(\mathcal{T}_h)$, and integrate over the element E to obtain

$$\int_E c\phi \frac{\partial p}{\partial t} v - \int_E \nabla \cdot \left((1 + cp) \frac{K}{\mu} \nabla p \right) v = \int_E (1 + cp) f v \quad (2.3)$$

The second integral can be rewritten using the divergence theorem:

$$-\int_E \nabla \cdot \left((1 + cp) \frac{K}{\mu} \nabla p \right) v = \int_E (1 + cp) \frac{K}{\mu} \nabla p \cdot \nabla v - \int_{\partial E} (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} v,$$

where \mathbf{n} is the outward facing normal of element E . Applying this change to equation (2.3) and summing over all elements $E \in \mathcal{T}_h$, we obtain

$$\begin{aligned} \sum_{E \in \mathcal{T}_h} \int_E c \phi \frac{\partial p}{\partial t} v + \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) \frac{K}{\mu} \nabla p \cdot \nabla v - \sum_{E \in \mathcal{T}_h} \int_{\partial E} (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} v \\ = \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) f v \end{aligned} \quad (2.4)$$

Sum over edges: The third sum in equation (2.4) can be rewritten by noting that every interior edge will be integrated over twice, as it will be a part of the boundary of exactly two elements in \mathcal{T}_h . Let Γ_h^0 denote the set of interior edges, and let $e \in \Gamma_h^0$. Let $E_{i_1}, E_{i_2} \in \mathcal{T}_h$ be such that $E_{i_1} \cap E_{i_2} = e$, and without loss of generality, assume that $i_1 < i_2$. Further, define \mathbf{n}_1 to be a unit normal vector which points from element E_{i_1} to E_{i_2} , and define $\mathbf{n}_2 = -\mathbf{n}_1$. With this notation, the following integral may be rewritten:

$$\begin{aligned} \int_e (1 + c(p)|_{E_1}) \frac{K}{\mu} (\nabla p)|_{E_1} \cdot \mathbf{n}_1 v|_{E_1} + \int_e (1 + c(p)|_{E_2}) \frac{K}{\mu} (\nabla p)|_{E_2} \cdot \mathbf{n}_2 v|_{E_2} \\ = \int_e \left(\left((1 + cp) \frac{K}{\mu} \nabla p v \right) \Big|_{E_1} - \left((1 + cp) \frac{K}{\mu} \nabla p v \right) \Big|_{E_2} \right) \cdot \mathbf{n}_1. \end{aligned} \quad (2.5)$$

For ease of notation, the jump function is introduced: for an edge $e = E_1 \cap E_2$, the jump of a function ψ along $e \in \Gamma_h^0$ is defined as

$$[\psi] = \psi|_{E_1} - \psi|_{E_2}$$

Utilizing the jump function, equation (2.5) may be rewritten more compactly as

$$\int_e \left[(1 + cp) \frac{K}{\mu} \nabla p v \right] \cdot \mathbf{n}_1$$

This holds for all $e \in \Gamma_h^0$. However, every edge along the boundary of the domain (that is, $e \in \Gamma_h \setminus \Gamma_h^0$) will only be integrated over once, since there is only a single element in \mathcal{T}_h which has e on its boundary. The jump function is modified to handle this case, and to allow for a more compact final equation. For an edge $e \in \Gamma_h \setminus \Gamma_h^0$, the jump of a function v along e is simply defined as

$$[\psi] = \psi$$

Therefore, the jump of a function ψ over any edge (i.e., $e \in \Gamma_h$) is given as

$$[\psi] = \begin{cases} \psi|_{E_{i_1}} - \psi|_{E_{i_2}} & \text{if } e \in \Gamma_h^0 \\ \psi & \text{if } e \in \Gamma_h \setminus \Gamma_h^0 \end{cases}$$

Although the second case may seem like a trivial inclusion, this definition allows the third sum in equation (2.4) to be written as:

$$\sum_{E \in \mathcal{T}_h} \int_{\partial E} (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} v = \sum_{e \in \Gamma_h} \int_e \left[(1 + cp) \frac{K}{\mu} \nabla p v \right] \cdot \mathbf{n} \quad (2.6)$$

where \mathbf{n} in the second integral is understood to be the unit normal pointing from E_{i_1} to E_{i_2} in the case of interior edges, and the outward pointing unit normal for boundary edges.

This sum can be decomposed further. Define the average of a function ψ along an edge $e \in \Gamma_h$ to be

$$\{\psi\} = \begin{cases} \frac{1}{2} (\psi|_{E_{i_1}} + \psi|_{E_{i_2}}) & \text{if } e \in \Gamma_h^0 \\ \psi & \text{if } e \in \Gamma_h \setminus \Gamma_h^0 \end{cases}$$

Then, for two functions a and b , note the following identity, for any elements E_1 and E_2 :

$$[ab] = a|_{E_1} b|_{E_1} - a|_{E_2} b|_{E_2}$$

$$\begin{aligned}
&= a|_{E_1} b|_{E_1} - \frac{1}{2}(a|_{E_1} b|_{E_2} - a|_{E_1} b|_{E_2} - a|_{E_2} b|_{E_1} + a|_{E_2} b|_{E_1}) - a|_{E_2} b|_{E_2} \\
&= \frac{1}{2}(a|_{E_1} b|_{E_1} - a|_{E_1} b|_{E_2} + a|_{E_2} b|_{E_1} - a|_{E_2} b|_{E_2}) + \frac{1}{2}(a|_{E_1} b|_{E_1} + a|_{E_1} b|_{E_2} - a|_{E_2} b|_{E_1} - a|_{E_2} b|_{E_2}) \\
&= \frac{1}{2}(a|_{E_1} + a|_{E_2})(b|_{E_1} - b|_{E_2}) + (a|_{E_1} - a|_{E_2}) \frac{1}{2}(b|_{E_1} + b|_{E_2}) = \{a\}[b] + [a]\{b\}
\end{aligned}$$

Using this identity, the sum in equation (2.6) can be rewritten as:

$$\begin{aligned}
&\sum_{e \in \Gamma_h} \int_e \left[(1 + cp) \frac{K}{\mu} \nabla p v \right] \cdot \mathbf{n} \\
&= \sum_{e \in \Gamma_h} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla p \right\} [v] \cdot \mathbf{n} + \sum_{e \in \Gamma_h} \int_e \left[(1 + cp) \frac{K}{\mu} \nabla p \right] \{v\} \cdot \mathbf{n}
\end{aligned} \tag{2.7}$$

The solution to equation (2.4), p , satisfies

$$\left[(1 + cp) \frac{k}{\mu} \nabla p \right] \cdot \mathbf{n} = 0 \text{ almost everywhere on } \Gamma_h^0,$$

Using this fact and the boundary condition in equation (2.2), the two sums in equation (2.7) can be reduced to

$$\begin{aligned}
&\sum_{e \in \Gamma_h} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla p \right\} [v] \cdot \mathbf{n} + \sum_{e \in \Gamma_h} \int_e \left[(1 + cp) \frac{K}{\mu} \nabla p \right] \{v\} \cdot \mathbf{n} \\
&= \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla p \right\} [v] \cdot \mathbf{n}.
\end{aligned}$$

This implies that equation (2.4) becomes

$$\begin{aligned}
&\sum_{E \in \mathcal{T}_h} \int_E c \phi \frac{\partial p}{\partial t} v + \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) \frac{K}{\mu} \nabla p \cdot \nabla v - \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} \right\} [v] \\
&= \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) f v
\end{aligned} \tag{2.8}$$

Symmetry terms: At it currently stands, the solution to equation (2.8) does not necessarily have a solution, and if it does, this solution is not necessarily unique. In order to ensure the existence and uniqueness of a solution for the DG method, both

symmetry and penalty terms must be added to the formulation. For this reason, I add the following terms to the left-hand side of the equation:

$$-\epsilon \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla v \cdot \mathbf{n} \right\} [p]$$

where ϵ is defined by the type of DG method being utilized:

$$\epsilon = \begin{cases} +1 & \text{Symmetric Interior Penalty Galerkin method (SIPG)} \\ -1 & \text{Nonsymmetric Interior Penalty Galerkin method (NIPG)} \\ 0 & \text{Incomplete Interior Penalty Galerkin method (IIPG)} \end{cases}$$

For my tests, I use the NIPG method, meaning that $\epsilon = -1$. Note that, since $[p] = 0$ almost everywhere over all interior edges, the terms being added are equal to 0. This means that consistency of the weak form is maintained when the symmetry terms are added to the left-hand side.

Penalty terms: As with the symmetry terms, I add the following sum to the left-hand side of equation (2.8):

$$\sum_{e \in \Gamma_h^0} \int_e \frac{\sigma}{|e|} [p][v],$$

in order to ensure existence and uniqueness of a solution. Again, note that the consistency of the original weak form is maintained due to regularity assumptions on p . In this notation, σ is a positive number, known as the penalty parameter, and $|e|$ is the length of edge e . The stable form of the discretization is given as

$$\begin{aligned} & \sum_{E \in \mathcal{T}_h} \int_E c\phi \frac{\partial p}{\partial t} v + \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) \frac{K}{\mu} \nabla p \cdot \nabla v \\ & - \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla p \cdot \mathbf{n} \right\} [v] + \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp) \frac{K}{\mu} \nabla v \cdot \mathbf{n} \right\} [p] \\ & + \sum_{e \in \Gamma_h^0} \int_e \frac{\sigma}{|e|} [p][v] = \sum_{E \in \mathcal{T}_h} \int_E (1 + cp) f v. \end{aligned} \quad (2.9)$$

Backward Euler discretization: Equation (2.9) will now be discretized with the backward Euler method. Let the pressure at the current time-step, $n + 1$, be denoted as p^{n+1} , and the pressure at the past time-step, n , be denoted as p^n . Further, denote the source term at the $n + 1$ time-step, $f(t_{n+1})$, as

$$f(t_{n+1}) = f^{n+1}.$$

The backward Euler discretization of equation (2.9) is

$$\begin{aligned} & \sum_{E \in \mathcal{T}_h} \int_E c\phi \frac{p^{n+1} - p^n}{\Delta t} v + \sum_{E \in \mathcal{T}_h} \int_E (1 + cp^{n+1}) \frac{K}{\mu} \nabla p^{n+1} \cdot \nabla v \\ & - \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp^{n+1}) \frac{K}{\mu} \nabla p^{n+1} \cdot \mathbf{n} \right\} [v] + \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp^{n+1}) \frac{K}{\mu} \nabla v \cdot \mathbf{n} \right\} [p^{n+1}] \\ & + \sum_{e \in \Gamma_h^0} \int_e \frac{\sigma}{|e|} [p^{n+1}][v] = \sum_{E \in \mathcal{T}_h} \int_E (1 + cp^{n+1}) f^{n+1} v. \end{aligned} \quad (2.10)$$

Although equation (2.10) is the true backward Euler discretization, implementing this would lead to a non-linear system of equations, meaning that an iterative method would be required at each time-step to obtain a solution. To avoid this, I time lag all of the $(1 + cp^{n+1})$ terms in equation (2.10) to obtain:

$$\begin{aligned} & \sum_{E \in \mathcal{T}_h} \int_E c\phi \frac{p^{n+1} - p^n}{\Delta t} v + \sum_{E \in \mathcal{T}_h} \int_E (1 + cp^n) \frac{K}{\mu} \nabla p^{n+1} \cdot \nabla v \\ & - \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp^n) \frac{K}{\mu} \nabla p^{n+1} \cdot \mathbf{n} \right\} [v] + \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp^n) \frac{K}{\mu} \nabla v \cdot \mathbf{n} \right\} [p^{n+1}] \\ & + \sum_{e \in \Gamma_h^0} \int_e \frac{\sigma}{|e|} [p^{n+1}][v] = \sum_{E \in \mathcal{T}_h} \int_E (1 + cp^n) f^{n+1} v. \end{aligned} \quad (2.11)$$

Final numerical scheme: Denote p_h^{n+1} as the numerical solution obtained by the DG discretization, at time-step $n + 1$. Moving all the terms with unknowns to the left-hand side of equation (2.11), all terms with known data to the right-hand side,

and multiplying by Δt , the final numerical scheme of equation (2.2) becomes

$$\begin{aligned}
& \sum_{E \in \mathcal{T}_h} \int_E c\phi p_h^{n+1} v + \Delta t \sum_{E \in \mathcal{T}_h} \int_E (1 + cp_h^n) \frac{K}{\mu} \nabla p_h^{n+1} \cdot \nabla v \\
& - \Delta t \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp_h^n) \frac{K}{\mu} \nabla p_h^{n+1} \cdot \mathbf{n} \right\} [v] + \Delta t \sum_{e \in \Gamma_h^0} \int_e \left\{ (1 + cp_h^n) \frac{K}{\mu} \nabla v \cdot \mathbf{n} \right\} [p_h^{n+1}] \\
& + \Delta t \sum_{e \in \Gamma_h^0} \int_e \frac{\sigma}{|e|} [p_h^{n+1}] [v] = \Delta t \sum_{E \in \mathcal{T}_h} \int_E (1 + cp_h^n) f^{n+1} v + \sum_{E \in \mathcal{T}_h} \int_E c\phi p_h^n v
\end{aligned} \tag{2.12}$$

2.3 ReservoirGrail: Equalization Rate

ReservoirGrail aims to simplify the oil reservoir model by introducing an artificial parameter, called equalization (EQ) rate, whose purpose is to replace the physical quantity K/μ , the permeability to viscosity ratio, present in the traditional model. One of the goals of my research is to determine if there is a relationship between the EQ rate and K/μ , and what the correlation is if it exists.

In order to determine this correlation, I first run the traditional model at fixed permeability values, between 10mD and 1000mD. I then run ReservoirGrail at varied EQ rates, from 0 to 1 in increments of 0.1, and save the results. I use three different norms to calculate the error between the ReservoirGrail and traditional simulator results: the ℓ_1 , ℓ_2 , and ℓ_∞ norms, which are defined as:

$$\begin{aligned}
\text{For } \mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \in \mathbb{R}^n : & \|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|, \\
& \|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^n v_i^2}, \\
& \|\mathbf{v}\|_\infty = \max_{1 \leq i \leq n} |v_i|.
\end{aligned}$$

To determine how well the results from the traditional model and ReservoirGrail match each other, I generate the ℓ_1 , ℓ_2 , and ℓ_∞ norm errors between the two sets of results. For every test performed, tables of the generated results can be found in appendix A. The “best match EQ rate” for each scenario is defined to be the EQ rate that minimizes one of the three errors measured; the norm to minimize depends on the scenario itself, and is determined on a case by case basis in Section 3.

2.4 Solution Conversion

The unknown in the PDE defined in Section 2.1 is pressure, p , which means the generated solution will show the pressure distribution across each element in the domain. However, ReservoirGrail generates solutions that depict the hydrocarbon pore volume (HPV) in each element of the mesh. The HPV value in a cell gives the amount of hydrocarbons present per square foot in the cell. In all of the tests I perform in this thesis, oil is the only hydrocarbon present, meaning that the HPV value is in units of barrels of oil per square foot, bbl/ft²; for more information on the unit “barrel”, see Appendix B. To compare the results of both the traditional model and ReservoirGrail, the solutions must be in the same units. I propose converting both solutions to units of pounds (lbs) of oil, which can then be easily converted into barrels if necessary.

Denote the mass of oil in an element $E \subset \mathbb{R}^2$ at time t_n by m_E^n . In order to convert pressure to mass with the traditional model, note the following mathematical definition for the mass in an element E :

$$m_E^n = D \int_E \rho(t_n) \, dV,$$

where ρ is the density of the material in question in units of lbs/ft³, and D is the

depth of the reservoir at element E in units of feet. Recall from Section 2.1 the ideal fluid density approximation:

$$\rho(p(t)) = \rho_{\text{ref}}(1 + cp(t)).$$

Therefore, calculating the mass in each element becomes equivalent to performing the calculation

$$m_E^n = D \int_E \rho_{\text{ref}}(1 + cp^n) dV \quad (2.13)$$

The parameter ρ_{ref} is the reference density, which is the density of the fluid at STP, and c is the compressibility of the fluid. The solution p is post-processed in this manner after solving the PDE system, to obtain a distribution of mass across the domain.

Converting HPV to mass with ReservoirGrail's results is more straightforward, and only requires the introduction of several conversion factors. ReservoirGrail produces many different files once it is executed, one of which being the "TransHPV" files. These files contain the solution output, in units of barrels of oil per square foot, at reservoir conditions. Let h be the mesh spacing in units of feet, ρ_{ref} be the reference density of the oil at STP, and note that the conversion factor from bbl to ft^3 is $5.614 \text{ ft}^3/\text{bbl}$. Then, dimensional analysis shows that

$$\left[\text{HPV } h^2 \rho_{\text{ref}} 5.614 \frac{\text{ft}^3}{\text{bbl}} \right] = \left[\frac{\text{bbl}}{\text{ft}^2} \text{ft}^2 \frac{\text{lbs}}{\text{ft}^3} \frac{\text{ft}^3}{\text{bbl}} \right] = [\text{lbs}].$$

Therefore, multiplying these parameters together will give a value in lbs of oil. However, since the HPV values are given in the setting of reservoir conditions, and the density and conversion factor are given at STP, a final conversion factor of

$$\frac{1}{1 - c\bar{p}} \quad (2.14)$$

is necessary, where \tilde{p} is defined as the difference in pressure in psi between the subsea reservoir and the sea-level platform that the oil is extracted to. Note that this conversion factor in equation (2.14) assumes that the temperature does not change from reservoir to standard conditions. Let p_w be the pressure at the subsea well head, and let p_s be the sea level pressure; then, \tilde{p} is defined as

$$\tilde{p} = p_w - p_s.$$

The pressure at sea level is given as 14.7 psi; therefore, this equation becomes

$$\tilde{p} = p_w - 14.7,$$

where p_w is a parameter that can be varied in both the traditional simulator and in ReservoirGrail.

Define HPV_E to be the reported HPV value in element E ; then, the final conversion from HPV to lbs of oil is given as

$$m_E = \frac{5.614 \text{ HPV}_E h^2 \rho_{\text{ref}}}{1 - c\tilde{p}}$$

After these two conversions, the results of both simulators are compared directly.

Chapter 3

Results

The purpose of this chapter is to describe the tests I ran to compare the two simulators, as well as to analyze and give an interpretation of the generated results. This includes graphs of the error versus equalization rate, heatmaps of the distribution of the error across the reservoir, and a localized 40 acre error analysis. I first layout the general parameters used in both simulators in Section 3.1, and then present the tests, their results, and my analysis as it pertains to the accuracy of ReservoirGrail in Section 3.2.

3.1 Test Setup

In Section 3.1.1, I explain the techniques I use to analyze the data generated from the different considered scenarios, including error calculations and interpretations. I then discuss the parameters in both reservoirs that are held constant throughout the simulations in Section 3.1.2, and give an explanation pertaining to their values.

3.1.1 Data Analysis

Once the results are obtained from both models for a given test scenario, I analyze the data in three different ways in order to provide a well-rounded and informative depiction of the results. These different analysis strategies I utilize are determined by what would be the most helpful way to visualize the differences between the two models, as well as by what features are important for industrial consideration.

The first technique I use is different between the first reservoir scenario (single and double producer) and the following scenarios (variable permeability and impermeable barriers). With the first scenario, I provide graphs showing how the best match EQ rate changes with the K/μ ratio, as well as showing how the ℓ_1 , ℓ_2 , and ℓ_∞ percent errors change with K/μ . With the other scenarios, I instead hold K/μ constant and show graphs of the EQ rate vs. the ℓ_1 , ℓ_2 , and ℓ_∞ percent errors. For the next data analysis technique, I provide heatmaps of the percent error across the domain for the best-match EQ rate's results, as well as a heatmap of error in terms of barrels of oil (e.g., bbl error), in order to provide a global view of the discrepancies between the two simulators.

These two tests are meant to compare the two models' results in a straightforward manner; however, the final method I use to analyze the data is a 40 acre test, which focuses on determining how accurate ReservoirGrail is in determining the oil remaining in a given area around the well-head(s). I calculate this value by summing the difference (without taking an absolute value) in the remaining oil between the two models in a 40 acre area around the well-head(s) (equivalently, in a circle with radius 750 feet centered at the well-head(s)).

3.1.2 Static Reservoir Variables

As most of the variables in the traditional and ReservoirGrail simulators do not change across tests, it will be most beneficial to list these variables once and refer back to them many times. Table 3.1 shows all of the variables used in the traditional model, Table 3.2 shows all of the variables used in ReservoirGrail, and Table 3.3 shows all of the variables shared in both models.

The specific values for μ , c , and p_{init} in Table 3.1 were chosen in order to represent

<u>Traditional Model Variables</u>	<u>Value</u>
ρ_{ref}	54.31 lbs/ft ³
μ	10 cP
c	$6.953932712 \times 10^{-6} \text{psi}^{-1}$
p_{init}	2250 psi
$f(t_{n+1})$	$-3.251103483 (1 + cp^n)^{-1} \text{month}^{-1}$

Table 3.1 : Variables and their values in the PDE model.

a realistic reservoir scenario. For more information on units and their realistic values, see Appendix B. Note that a constant source function does not imply a constant production rate, it implies a constant change in pressure across the domain: for this reason, the source function given in Table 3.1 is time dependent. The specific values in the source function were obtained in order to replicate the constant production rate defined in Table 3.3.

<u>ReservoirGrail Variables</u>	<u>Value</u>
B_o	1
$\frac{1}{B_g}$	1
S_o	1
S_g	0

Table 3.2 : Variables and their values in the agent-based model.

In Table 3.2, B_o is the oil formation factor, and B_g is the gas formation factor. These values represent the shrinkage/expansion factors from reservoir to surface

conditions of oil and gas, respectively, due to both the compressibility of the two hydrocarbons, as well as the amount of gas that is dissolved into the oil at reservoir conditions. As the work done in this thesis assumes that there is no gas present in the modeled reservoir, the value of B_g is irrelevant (and is therefore set to be 1), and furthermore, the oil formation factor only takes into account the amount at which oil expands as it is brought from reservoir to surface conditions due to its compressibility. However, as discussed in Section 2.4, the conversion factor given in equation (2.14) takes care of this consideration. Therefore, the value of $B_o = 1$ is given to model oil that does not expand as it moves from reservoir to surface conditions, and this expansion factor is taken care of after the simulation is complete with equation (2.14). The variable S_o is the initial oil saturation in the reservoir, and is set to 1 in all of the tests in order to model a reservoir full of oil, with no water or gas. Similarly, the variable S_g represents the initial gas saturation of the reservoir, and is set to 0 in the tests to model a reservoir with no gas present.

The production rate in Table 3.3 was obtained by assuming that 15% of the reservoir's total available hydrocarbons are extracted in its first 10 years of production, or equivalently, an extraction rate of 0.125% of the reservoir's initial hydrocarbons every month. This corresponds to the production rate in Table 3.3, under the assumption of single phase flow.

Furthermore, a porosity of 1 was used throughout the simulations. For an explanation and realistic values for ϕ , see Appendix B.

3.2 Computational Tests

I now go through the different reservoir scenarios that I tested, analyze the data as described in Section 3.1.1, and provide an interpretation as to the degree of accuracy of

<u>Common Variables</u>	<u>Value</u>
ϕ	1
p_w	2250 psi
D	10 ft
Length of simulation	11 months
Time-step	1 month
Domain	$(0, 2550) \times (0, 2550)$ feet
Grid spacing	50 ft
Cell layout	51×51
Production rate	14476.608 barrels of oil/month

Table 3.3 : Variables and their values that are common to both models.

the agent-based model. In Section 3.2.1, I test multiple reservoirs with varying homogeneous permeability values in the traditional model against reservoirs with varying EQ rates in ReservoirGrail, in order to determine a correlation between permeability and EQ rate. These tests are performed with both a single production well present, as well as two production wells. From these tests, I begin testing more complex reservoir scenarios: in Section 3.2.2, I test a reservoir with a permeability map that changes quickly and to a large degree near the production well, and in Sections 3.2.3 and 3.2.4, I consider reservoirs with both impermeable barriers and variable permeability across their domain. In Section 3.2.5, I consider single producer reservoirs with small homogeneous permeability and EQ rate values, in order to explore the effect that these small parameters have on the error between the two models. Finally, in Section 3.2.6 I consider the effects of computing solutions of two different reservoir setups on

a refined mesh in the traditional model, and comparing them to the output generated by ReservoirGrail on a standard mesh.

3.2.1 Single and Double Producers

Two simple reservoir scenarios were tested with both models, with the viscosity μ being held constant, in order to determine a correlation between the EQ rate and K/μ ratio, as well as to determine the accuracy of ReservoirGrail in comparison to the traditional model. Since μ is constant in these tests, determining a correlation between EQ rate and the permeability to viscosity ratio is equivalent to finding a correlation between the EQ rate and permeability, K . The tests in this section differed by the number and placement of production oil wells in the domain; figures 3.1 and 3.2 depict these two scenarios.

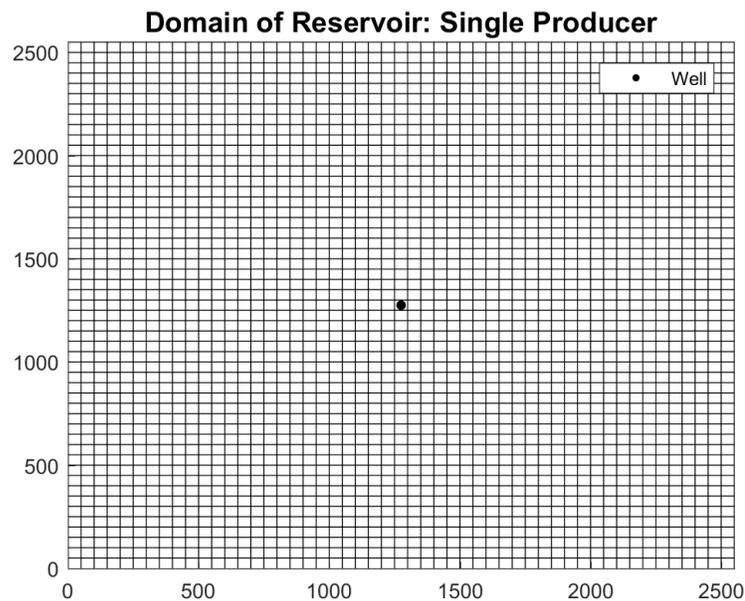


Figure 3.1 : Domain of oil reservoir for single producer test. The black dot marks the location of the production oil well.

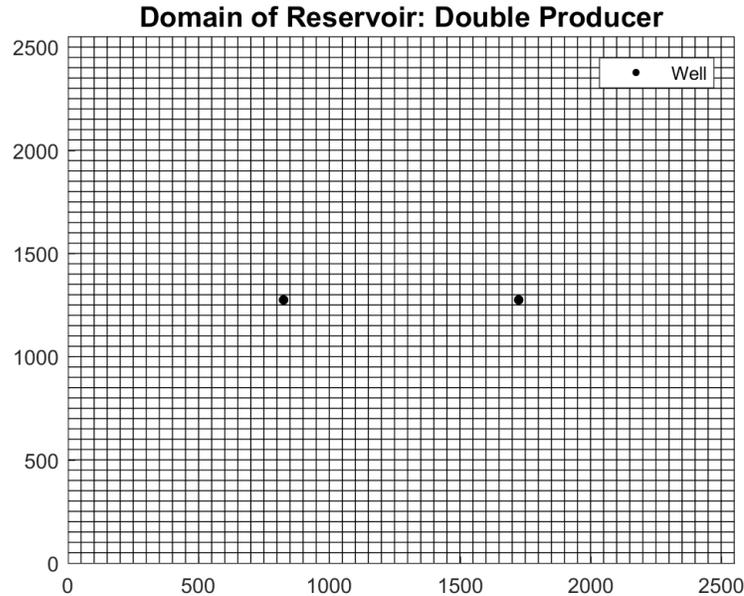


Figure 3.2 : Domain of oil reservoir for double producer test. The black dots mark the locations of the two production wells.

With the traditional model, permeability values of 10, 20, 30, 40, 50, 100, 200, 500, and 1000 mD were tested. For these simple reservoir setups with no additional features, I decided that minimizing the global error (hence, the ℓ_2 error) over the entire domain is more important than minimizing either of the other two errors. Therefore, for the single and double producer tests, I define the “best match EQ rate” to be the EQ value that minimizes the ℓ_2 error. Tables depicting the error values correlating to each EQ rate for each permeability value can be found in appendix A; specifically, tables A.1 through A.9 for the single producer tests, and tables A.10 through A.18 for the double producer tests.

Figure 3.3 shows the permeability vs. the ℓ_2 error (in percentage form) between the two models for the single-production scenario, and figure 3.4 shows the permeability

vs. its best match equalization rate for the single-production scenario.

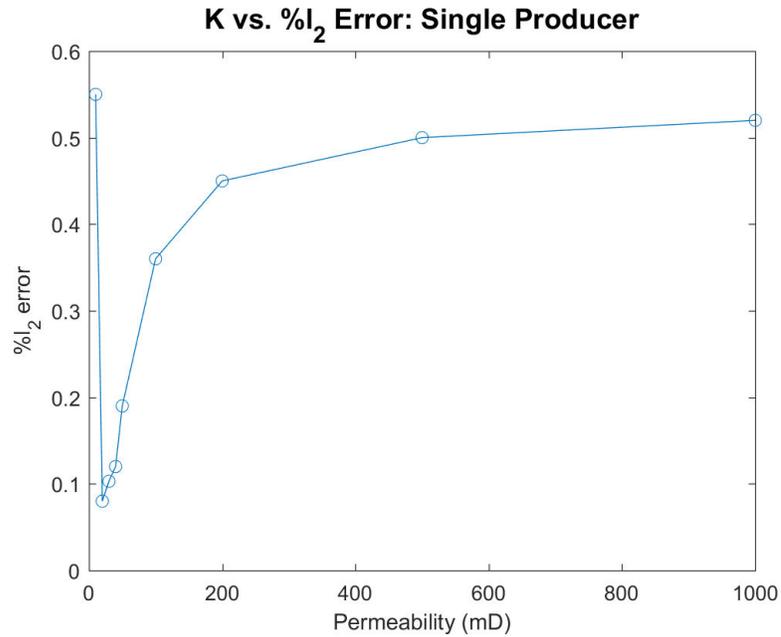


Figure 3.3 : Permeability value vs. l_2 error, in percentage, for a single production well scenario. Note that the first data point is higher than the rest, and that the error monotonically increases after the ratio climbs above 40mD. See tables A.1 through A.9.

Note how the error is low for permeability values of 20, 30 and 40mD in figure 3.3, and starts to grow monotonically after that. This behavior can be understood by considering figure 3.4, which shows that the best match EQ rate plateaus at a value of 1 at and after the reservoir tests with a permeability of 40mD and above. The reason for this sharp cutoff is that ReservoirGrail is programmed to not accept EQ rates above 1; see Section 1.2.3 for more information about how the EQ rate influences ReservoirGrail's simulations.

Further note in figure 3.3 that while all of the error is less than 0.6% for the

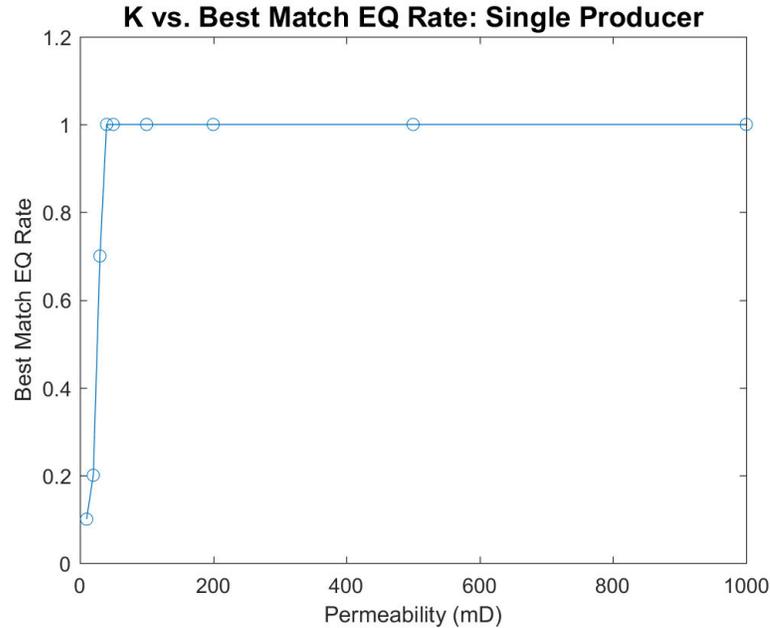


Figure 3.4 : Permeability value vs. best match EQ rate, for a single production well scenario. Note that the best match EQ rate plateaus after the permeability climbs above 40mD. See tables A.1 through A.9.

single producer test, a permeability value of 10mD gives an error that is relatively high, in comparison to the other permeability values. It is possible that this point is an outlier, and the high relative error is due to the coarse steps in EQ rate and permeability that were taken for the reservoir tests. I explore this idea further in Section 3.2.5 by “zooming in” on permeability values close to 10mD and EQ rates close to 0.1, taking smaller steps in both parameters to attempt to explain this large relative jump in error observed in figure 3.3.

Figure 3.5 shows the ℓ_1 error in percentage form, and figure 3.6 shows ℓ_∞ error in percentage form, both for the single-production scenario. The same features are apparent when the error is measured in these norms as well.

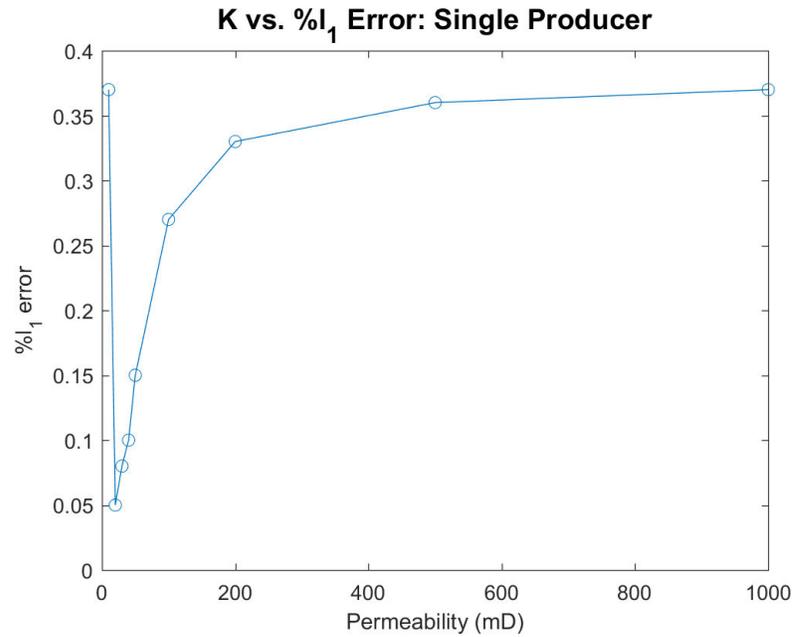


Figure 3.5 : Permeability value vs. l_1 error, in percentage, for a single production well scenario. Same behavior as depicted in figure 3.3. See tables A.1 through A.9.

The second scenario tested was one with two production wells, distributed across the domain as in figure 3.2. All other variables remain the same, except that the production rate for each well is half of that in the first test, in order to maintain the same global production rate between the two tests.

Figure 3.7 shows the ratio vs. the l_2 error (in percentage form) between the two models for the double-production scenario, and figure 3.8 shows the ratio vs. its best match equalization rate for the double-production scenario. The same features are present in these figures as in the single-production test: there is a relatively high error for the permeability value of 10mD, and the best match equalization rate plateaus at 1.0 after a permeability value of 30mD, leading to monotonically increasing error. Again, the same features can be seen in the l_1 error plot, figure 3.9, and in the l_∞

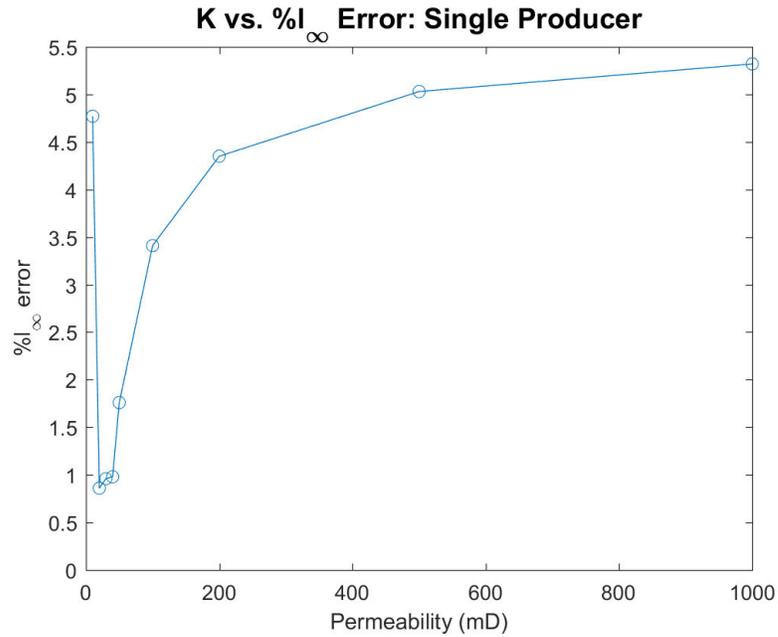


Figure 3.6 : Permeability value vs. l_∞ error, in percentage, for a single production well scenario. Same behavior as depicted in figure 3.3. See tables A.1 through A.9.

error plot, figure 3.10.

In addition to viewing the percent l_1 , l_2 , and l_∞ errors and their corresponding EQ rates, I plot the global error for both the single and double producer tests with a permeability of $K = 30\text{mD}$ (and corresponding EQ rate of 0.7 for the single producer test, and 1.0 for the double producer test), in terms of percent error and bbl error. This permeability was chosen as it is in the range of realistic values for oil reservoirs.

For the single producer test, figure 3.11 shows the percent error in the reservoir, and figure 3.12 shows error in units of barrels of oil.

Similarly, for the double producer test, figure 3.13 shows the percent error in the reservoir, and figure 3.14 shows the error in units of barrels of oil.

As discussed in Section 3.1.1, another metric I use to determine the difference

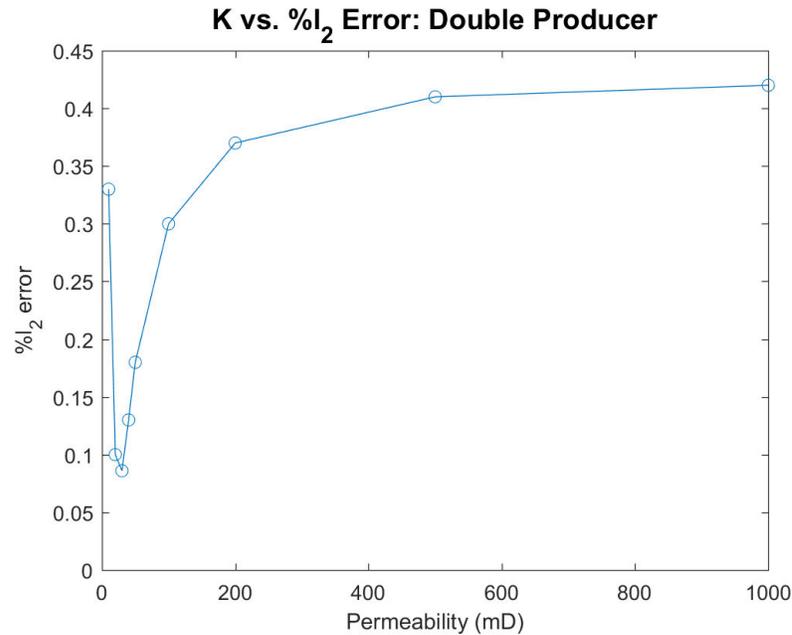


Figure 3.7 : Permeability value vs. l_2 error, in percentage, for a double production well scenario. Note that the first data point is higher than the rest, and that the error monotonically increases after the permeability climbs above 30mD. See tables A.10 through A.18.

between the two reservoir models is the 40-acre test. I calculate the difference in remaining oil between ReservoirGrail and the Darcy simulator, and sum up over cells that are within a circle with a radius of 750 feet. Figure 3.15 shows in yellow the cells that are considered in the single producer test, and figure 3.16 shows in yellow the cells that are considered in the double producer test.

When the difference is calculated as described above, Table 3.4 shows that ReservoirGrail reports less remaining oil within 40 acres than the traditional model, by at most 0.61% for permeability values at and above 20 mD. However, for a reservoir with a permeability of 10 mD, ReservoirGrail reports more remaining oil within 40

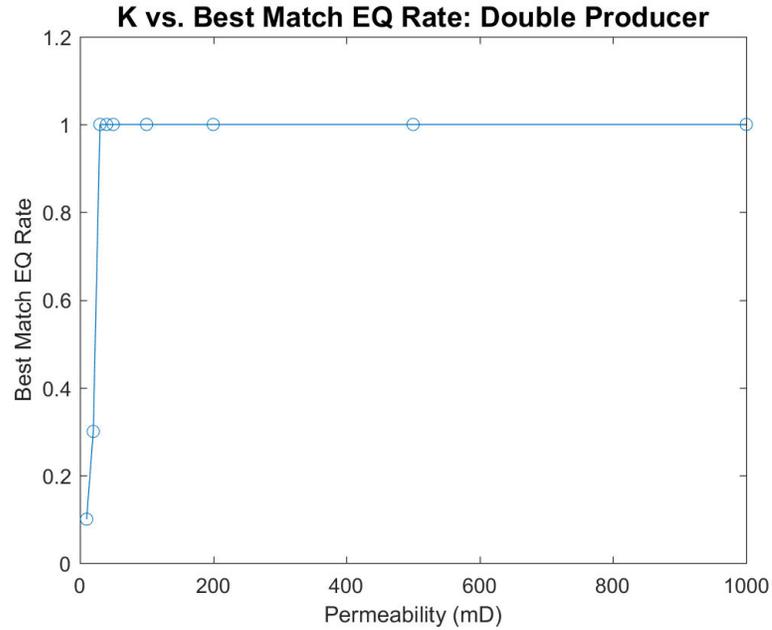


Figure 3.8 : Permeability value vs. best match EQ rate, for a double production well scenario. Note that the best match EQ rate plateaus after the permeability climbs above 30mD. See tables A.10 through A.18.

acres than the traditional model by 0.58%.

Similarly to Table 3.4, Table 3.5 show that ReservoirGrail underestimates the amount of remaining oil within 40 acres of either of the two wells by at most 0.37% for permeability values at and above 20 mD, and overestimates the amount of remaining oil by 0.23% for a reservoir with permeability of 10 mD.

3.2.2 Varied Permeability

Section 3.2.1 shows that for simple reservoir setups, there is a very low difference between the results produced by ReservoirGrail and the Darcy simulator. The purpose of this next test and the tests in Sections 3.2.3 and 3.2.4 are to start stressing

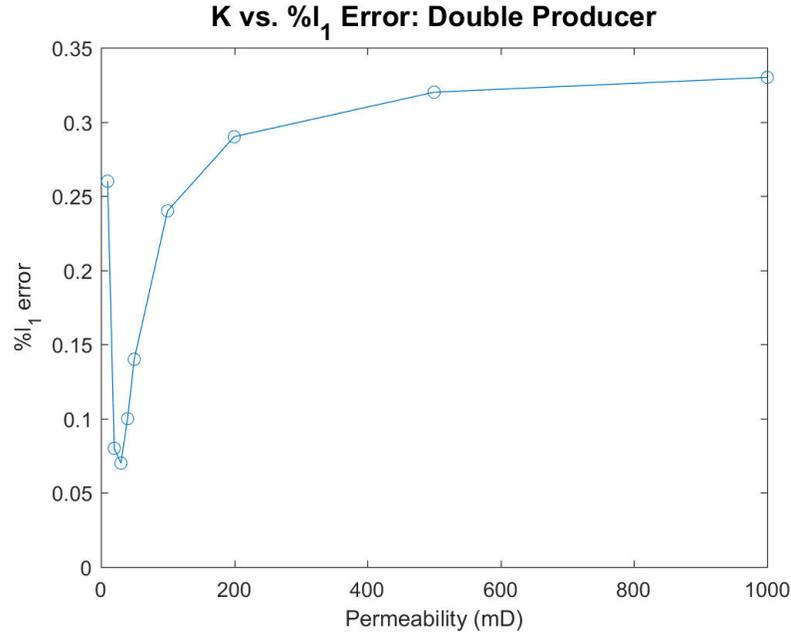


Figure 3.9 : Permeability value vs. l_1 error, in percentage, for a double production well scenario. Same behavior as depicted in figure 3.7. See tables A.10 through A.18.

the simulators by adding in features that are common in realistic oil reservoirs. The varied permeability reservoir has a well centered in the domain as before in the single producer test, but now includes a permeability profile that changes across the domain. Figure 3.17 shows this domain: the green cells represent areas with a permeability of 200 mD, and white cells represent areas with a permeability of 10 mD.

Other than the permeability and equalization rate, all of the variables used in both the traditional model and ReservoirGrail are listed in tables 3.1, 3.2, and 3.3.

For this setup, minimizing the local error (hence, the l_∞ error) was instead determined to be the appropriate metric to analyze the accuracy of ReservoirGrail, in order to control the error along the permeability interface and around the well-head. An equalization rate of 0.1 in ReservoirGrail was determined to give the closest results;

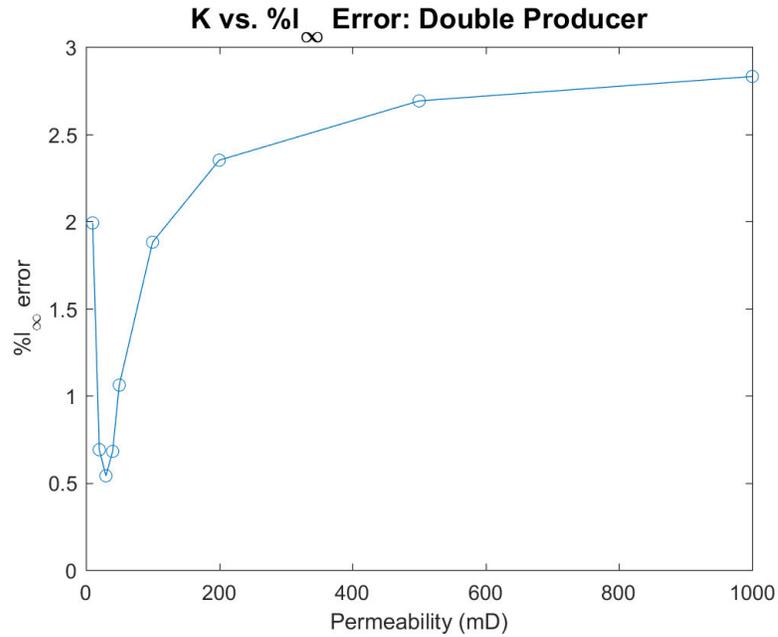


Figure 3.10 : Permeability value vs. l_{∞} error, in percentage, for a double production well scenario. Same behavior as depicted in figure 3.7. See tables A.10 through A.18.

see Table A.19. Table 3.6 shows the l_{∞} , l_2 , and l_1 norm errors in percentage form of the varied permeability test, figure 3.18 shows the error “heat-map” in percent error, and figure 3.19 shows the same heat-map in terms of barrels of oil.

Although the errors in this test are higher than in the single and double producer tests in Section 3.2.1, this is to be expected, as the overall complexity of the reservoir has increased by considering varied permeability values across the domain. This being said, even along the interface of the permeability discontinuity where one would expect a larger discrepancy to occur, the difference is still extremely low (between 1%-2% difference).

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 0.1. Table 3.7

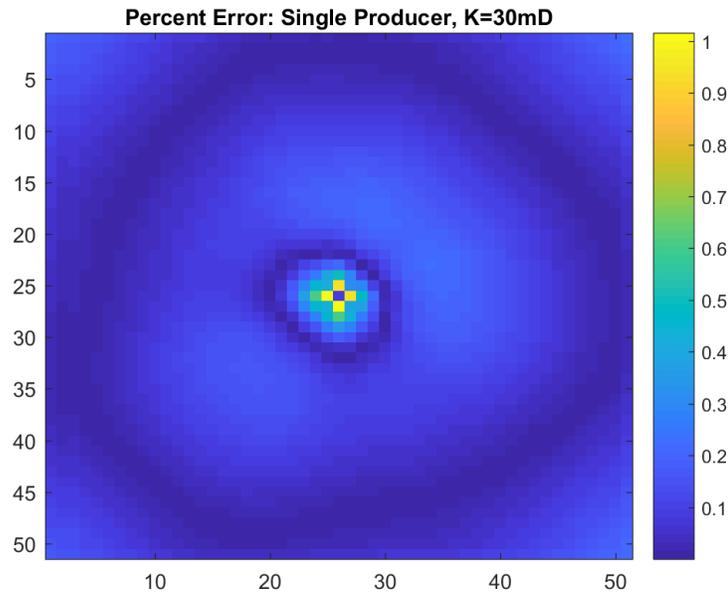


Figure 3.11 : Cell by cell percent error of the single producer test.

shows that the discrepancy between the two simulators for this analysis technique is approximately 0.01%.

3.2.3 Impermeable Barrier with Single Producer

Another feature that consistently show up in oil reservoirs are impermeable barriers, which no fluid can flow through. This thesis will compare results between the two simulators when such a barrier is introduced to the domain. I first show several sets of results for domains with a single barrier present to the left of the well, vary the permeability values uniformly across the domain. Next, I depict the results of a scenario with two such barriers present with a single oil well with homogeneous permeability. Finally, I present two sets of results depicting a domain with two barriers, two wells, and varied permeability; in the first test, the second barrier is to

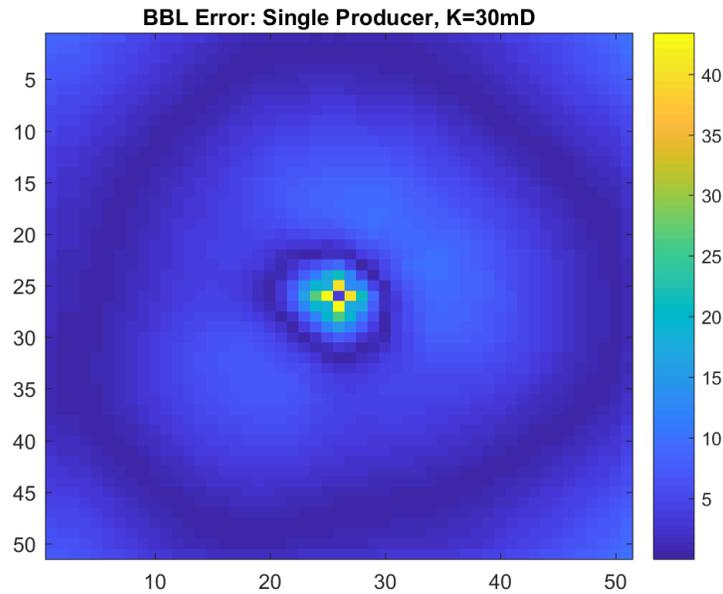


Figure 3.12 : Cell by cell barrels of oil error of the single producer test.

the left of both production wells, and in the second test, this barrier is moved so that it is splitting in between the two wells.

The way that ReservoirGrail models an impermeable barrier is by considering a line between specified points in the domain, and severing the connections between nodes that this line passes between, ensuring that the agents can no longer move freely between the nodes; see figure 3.20. In general, this allows for the modeling of one dimensional barriers in ReservoirGrail. The traditional model being utilized, however, has difficulty modeling one dimensional barriers; as such, only two dimensional barriers will be considered in this thesis.

In order to model a two dimensional barrier in ReservoirGrail, the specified line must pass directly through the nodes, which severs connections to that node from all of its neighbors; see figure 3.21. Analogously, modeling a two dimensional barrier in the

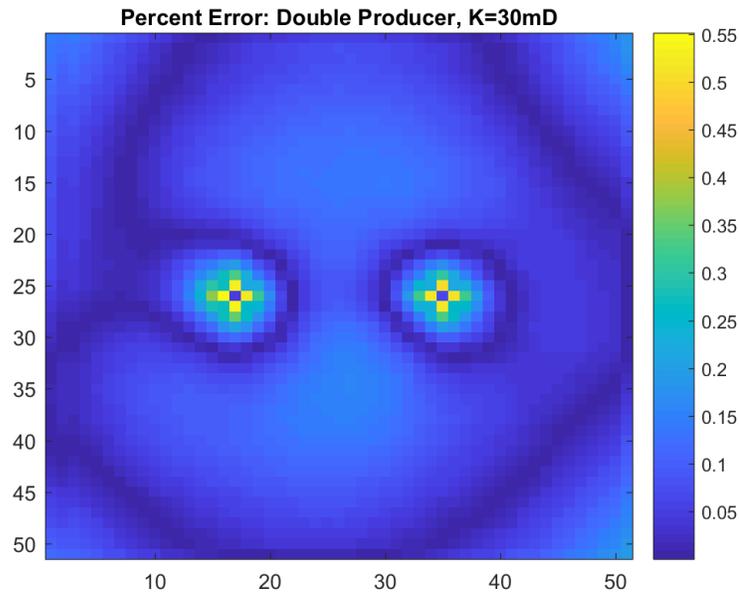


Figure 3.13 : Cell by cell percent error of the double producer test.

traditional model consists of removing elements from the domain in consideration, and assigning no-flow (that is, 0 Neumann) boundary conditions to the inner boundary.

Figure 3.22 shows the domain of this test, where the red elements represent the location of the barrier.

Three different permeability values for the reservoir were considered and tested: 30, 50, and 100 mD. The remaining variables are listed in tables 3.1, 3.2, and 3.3. For this setup, minimizing the local error (hence, the ℓ_∞ error) was determined to be the appropriate metric to analyze the accuracy of ReservoirGrail, in order to control the error along the impermeable barrier and around the well-head. The different permeability values are considered independently, in the three different sections below.

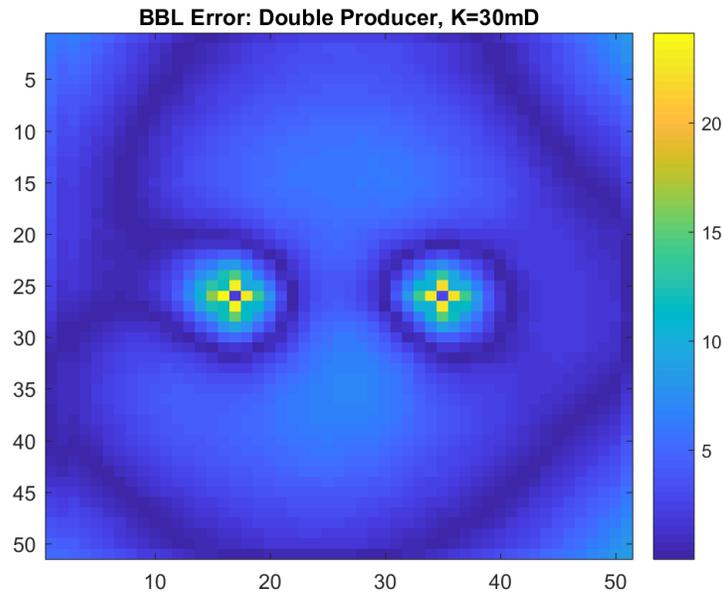


Figure 3.14 : Cell by cell barrels of oil error of the double producer test.

Permeability: 30mD

The results for the single barrier, 30 mD test are now shown.

The permeability of 30 mD corresponded to a best-match equalization rate of 0.4; see Table A.20. Table 3.8 shows the ℓ_∞ , ℓ_2 , and ℓ_1 norm errors in percentage form. Further, figure 3.23 shows the error heat map in terms of percent error, and figure 3.24 shows the error heat map in terms of barrels of oil. Note that the error is highest around the well in the center (which is to be expected), and is relatively low around the barrier.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 0.4. Table 3.9 shows that the two models differ by 0.11% in terms of total oil remaining, within 40 acres of the well.

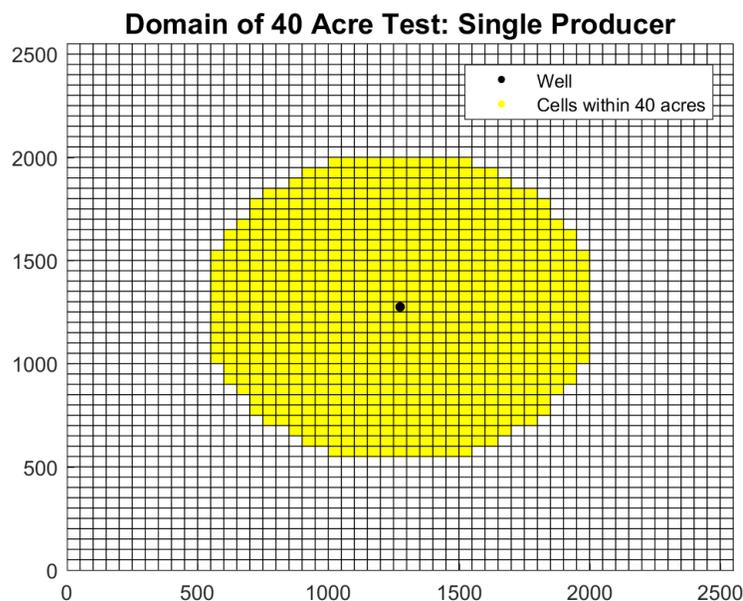


Figure 3.15 : Domain of 40 acre test with a single well. An area of 40 acres corresponds to a circle with radius 750 feet.

Permeability: 50mD

The results for the single barrier, 50 mD test are now shown.

The permeability of 50 mD corresponded to a best-match equalization rate of 1.0; see Table A.21 for more details. Table 3.10 shows the ℓ_∞ , ℓ_2 , and ℓ_1 norm errors in percentage form. Further, figure 3.25 shows the error heat map in terms of percent error, and figure 3.26 shows the error heat map in terms of barrels of oil. Note that the error is highest around the well in the center (which is to be expected), and is relatively low around the barrier. It is worth mentioning that although the maximum error increases from the 30mD case by approximately 0.7%, this is localized to near the well head; the error across the rest of the domain is still on the same order as before.

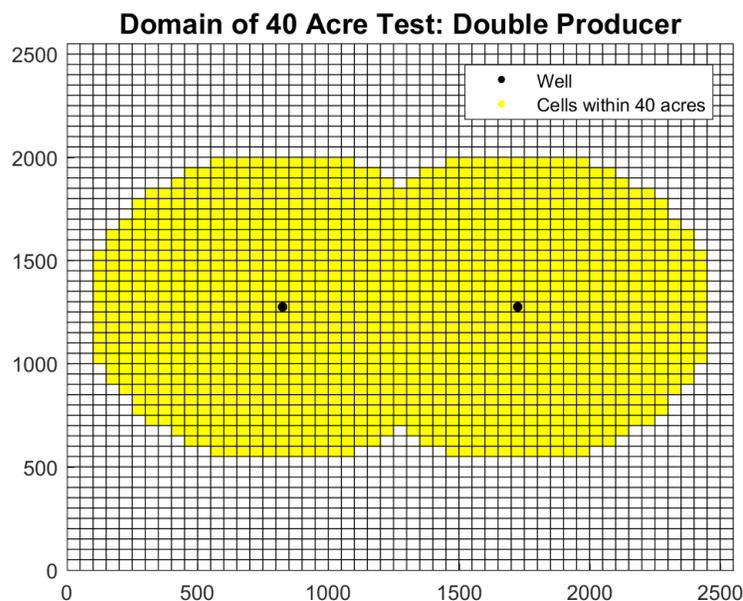


Figure 3.16 : Domain of 40 acre test with a single well. An area of 40 acres corresponds to a circle with radius 750 feet.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 1.0. Table 3.11 shows that the two models differ by 0.21% in terms of total oil remaining, within 40 acres of the well.

Permeability: 100mD

The results for the single barrier, 100 mD test are now shown.

The permeability of 100 mD corresponded to a best-match equalization rate of 1.0; see Table A.22 for more details. Table 3.12 shows the l_∞ , l_2 , and l_1 norm errors in percentage form. Further, figure 3.27 shows the error heat map in terms of percent error, and figure 3.28 shows the error heat map in terms of barrels of oil. Note that

<u>K(mD)</u>	<u>EQ</u>	<u>Difference (bbl)</u>	<u>Avg Diff (bbl/cell)</u>	<u>Difference (%)</u>
10	0.1	-20687	-29.7	0.58%
20	0.2	2302	3.3	0.06%
30	0.7	2827	4.1	0.08%
40	1.0	5135	7.4	0.14%
50	1.0	8664	12.4	0.24%
100	1.0	15782	22.6	0.43%
200	1.0	19349	27.8	0.53%
500	1.0	21491	30.8	0.59%
1000	1.0	22205	31.9	0.61%

Table 3.4 : The difference between Darcy’s law simulator and ReservoirGrail, for a single well. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

the error is highest around the well in the center (which is to be expected), and is relatively low around the barrier, which is analogous to the results of the 30mD and 50mD tests.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 1.0. Table 3.13 shows that the two models differ by 0.48% in terms of total oil remaining, within 40 acres of the well. While this is higher than both the 30mD and 50mD test, it is still extremely low.

<u>K(mD)</u>	<u>EQ</u>	<u>Difference (bbl)</u>	<u>Avg Diff (bbl/cell)</u>	<u>Difference (%)</u>
10	0.1	-12020	-10.0	0.23%
20	0.3	2714	2.3	0.05%
30	1.0	1939	1.6	0.04%
40	1.0	6430	5.0	0.12%
50	1.0	9186	7.7	0.17%
100	1.0	14763	12.3	0.28%
200	1.0	17562	14.7	0.33%
500	1.0	19245	16.1	0.36%
1000	1.0	19806	16.5	0.37%

Table 3.5 : The difference between Darcy’s law simulator and ReservoirGrail, for two wells. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

<u>Norm</u>	<u>Percent Error</u>
∞	3.52%
2	1.21%
1	0.94%

Table 3.6 : Percent errors for the varied permeability test. Note that while the values are slightly higher than in Section 3.2.1, they are still relatively low.

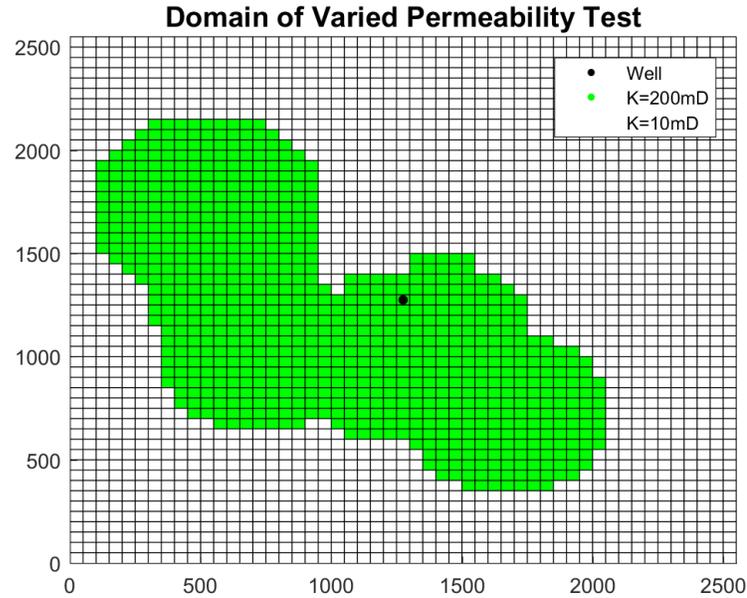


Figure 3.17 : The domain for the varied permeability test. The green area has a permeability of 200mD, while the white area has a permeability of 10mD.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
263	0.38	0.01%

Table 3.7 : The difference between the traditional and ReservoirGrail simulators, for the variable permeability test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

3.2.4 Double Impermeable Barrier

I now turn my attention to domains with two impermeable barriers present. I first consider a scenario with a single producer with barriers the the left and top of the production well and homogenous permeability, then consider two cases with two pro-

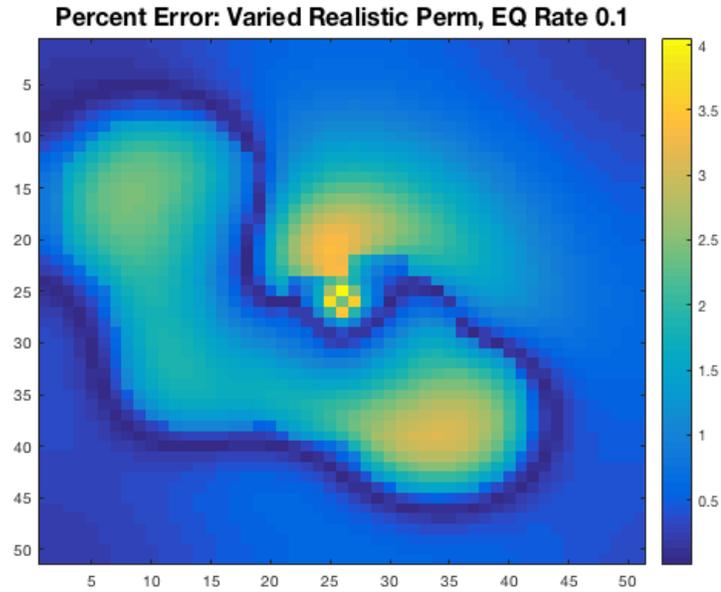


Figure 3.18 : Cell by cell percent error of varied permeability test.

<u>Norm</u>	<u>Percent Error</u>
∞	0.66%
2	0.15%
1	0.13%

Table 3.8 : Percent errors for the single barrier, 30 mD test. Note that while the values are slightly higher than in Section 3.2.1, they are still relatively low.

duction wells and variable permeability, one of which has the same barrier setup as the single producer test, and one with a barrier present in between the two production wells.

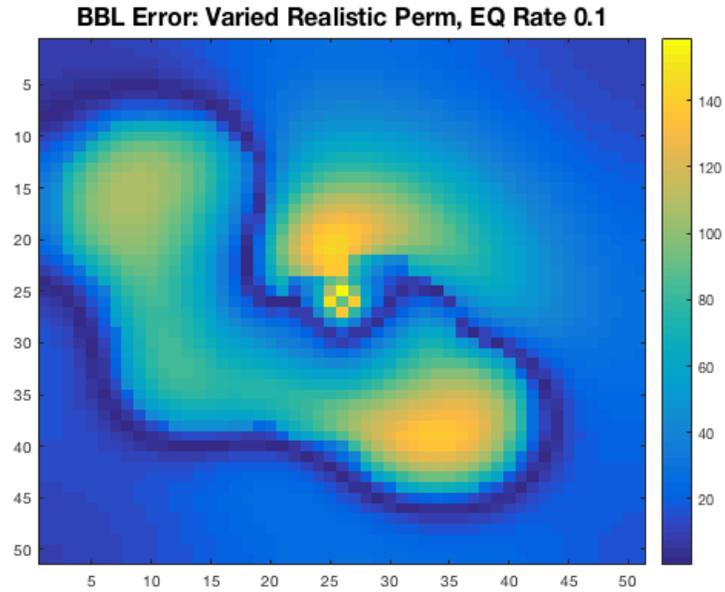


Figure 3.19 : Cell by cell barrels of oil error of varied permeability test.

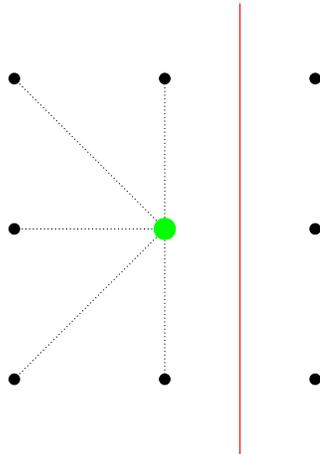


Figure 3.20 : Connectivity between the large, green central cell and its neighbors, with a red barrier severing some of the original connections. In this example, ReservoirGrail models the barrier as a one dimensional object.

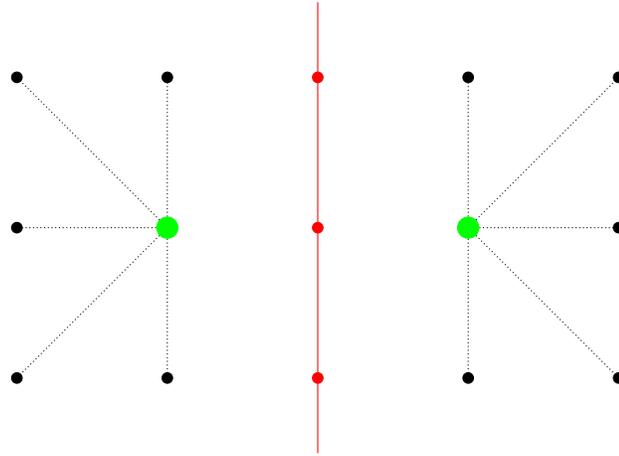


Figure 3.21 : Connectivity between the two large green cells and their neighbors, with a barrier severing some of their connections. In this example, ReservoirGrail models the barrier as a two dimensional object, as the red cells in the middle are essentially removed from the simulation.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
3859	5.62	0.11%

Table 3.9 : The difference between the traditional and ReservoirGrail simulators, for the single barrier, 30 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

Single Producer

Figure 3.29 shows the domain of this test, where the red elements represent the location of the barrier. A permeability value of 30 mD was selected for this test. The remaining variables are listed in tables 3.1, 3.2, and 3.3. As in the previous impermeable barrier tests, a high relative error is observed around the barrier itself,

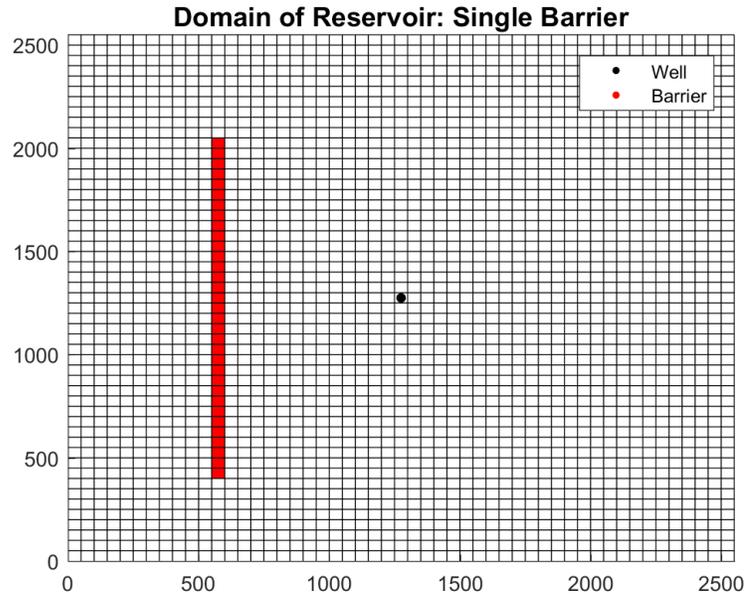


Figure 3.22 : The domain for the single barrier test. Red cells represent the location of the barrier.

<u>Norm</u>	<u>Percent Error</u>
∞	1.35%
2	0.22%
1	0.19%

Table 3.10 : Percent errors for the single barrier, 50 mD test.

in every EQ rate comparison. Therefore, minimizing the error along the barrier (hence, the ℓ_∞ norm) is determined to be the metric for this reservoir test.

This test setup corresponded to a best-match equalization rate of 0.3; see Table A.23 for additional details. Table 3.14 shows the ℓ_∞ , ℓ_2 , and ℓ_1 norm errors in percentage form. Further, figure 3.30 shows the error heat map in terms of percent

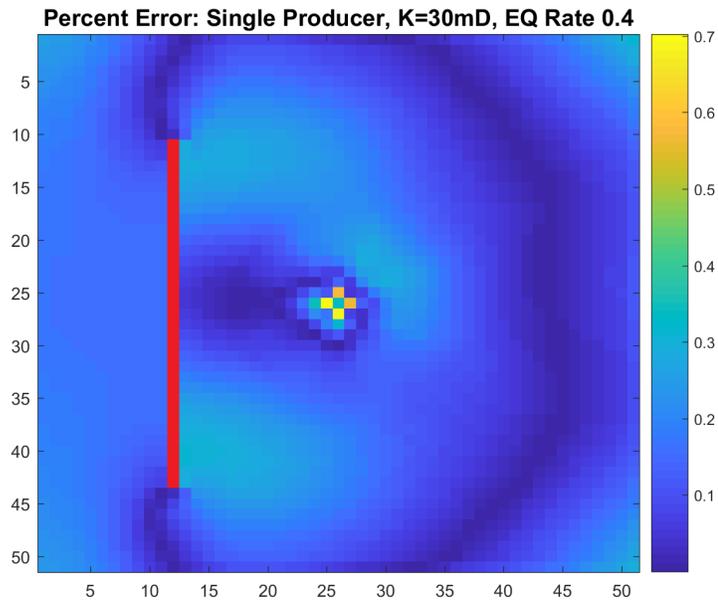


Figure 3.23 : Cell by cell percent error of the single barrier, 30 mD test.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
7543	10.99	0.21%

Table 3.11 : The difference between the traditional and ReservoirGrail simulators, for the single barrier, 50 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

error, and figure 3.31 shows the error heat map in terms of barrels of oil. Note that the areas of highest error in the domain are near the well in the center and along both of the barriers, which is to be expected.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 0.3. Table 3.15 shows that the two models differ by 0.05% in terms of total oil remaining, within 40 acres

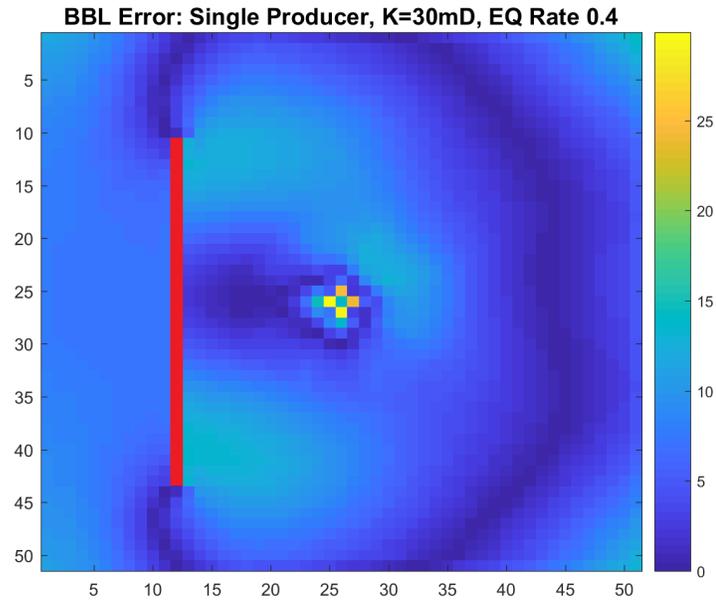


Figure 3.24 : Cell by cell barrels of oil error of the single barrier, 30 mD test.

<u>Norm</u>	<u>Percent Error</u>
∞	3.12%
2	0.44%
1	0.36%

Table 3.12 : Percent errors for the single barrier, 100 mD test.

of the well.

Double Producer, with Left Barrier and Varied Permeability

Figure 3.32 shows the domain of this test. Varied permeability values of 10 mD, 100 mD, and 1000 mD were used in this simulation, again shown in figure 3.32. The remaining variables are listed in tables 3.1, 3.2, and 3.3. As in the previous

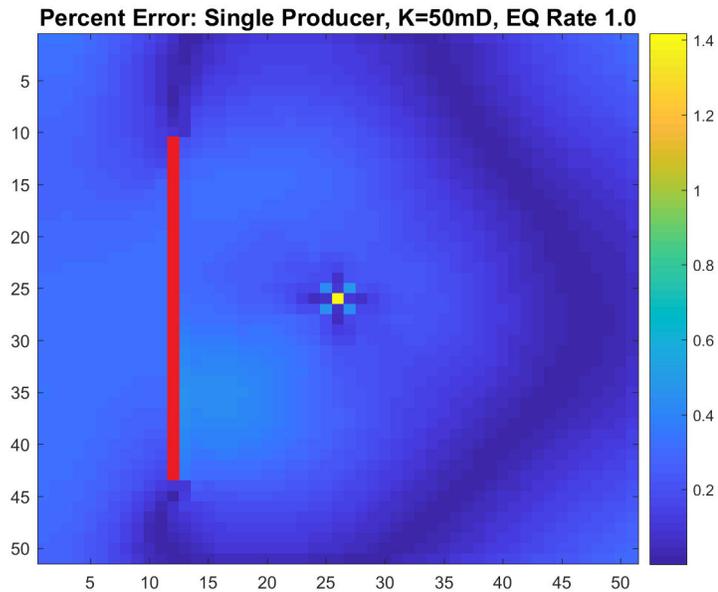


Figure 3.25 : Cell by cell percent error of the single barrier, 50 mD test.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
17331	25.26	0.48%

Table 3.13 : The difference between the traditional and ReservoirGrail simulators, for the single barrier, 100 mD test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

impermeable barrier tests, a high relative error is observed around the barriers, in every EQ rate comparison. Therefore, minimizing the error along the barrier (hence, the ℓ_∞ norm) is determined to be the metric for this reservoir test.

This test setup corresponded to a best-match equalization rate of 0.3; see Table A.24 for additional details. Table 3.16 shows the ℓ_∞ , ℓ_2 , and ℓ_1 norm errors in percentage form. Further, figure 3.33 shows the error heat map in terms of percent

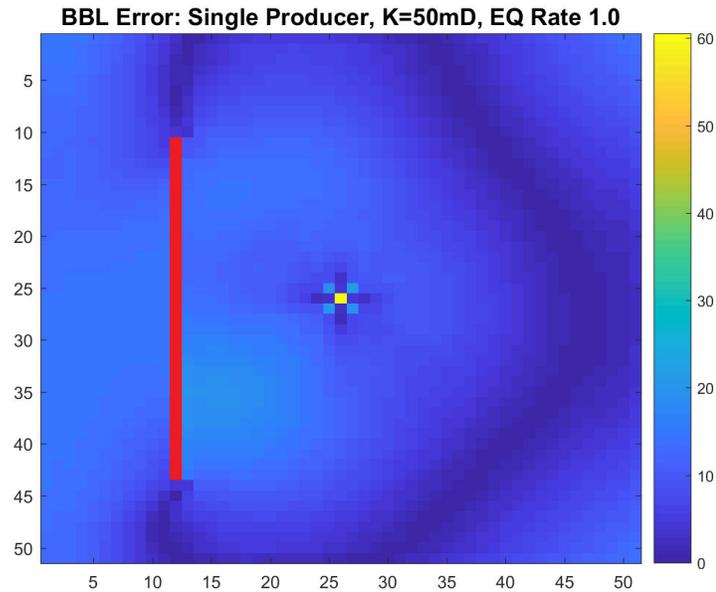


Figure 3.26 : Cell by cell barrels of oil error of the single barrier, 50 mD test.

<u>Norm</u>	<u>Percent Error</u>
∞	0.64%
2	0.19%
1	0.16%

Table 3.14 : Percent errors for the double barrier, single producer test.

error, and figure 3.34 shows the error heat map in terms of barrels of oil. Note that the areas of highest error in the domain are near the production wells, as is to be expected.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 0.3. Table 3.17 shows that the two models differ by 0.18% in terms of total oil remaining, within 40 acres

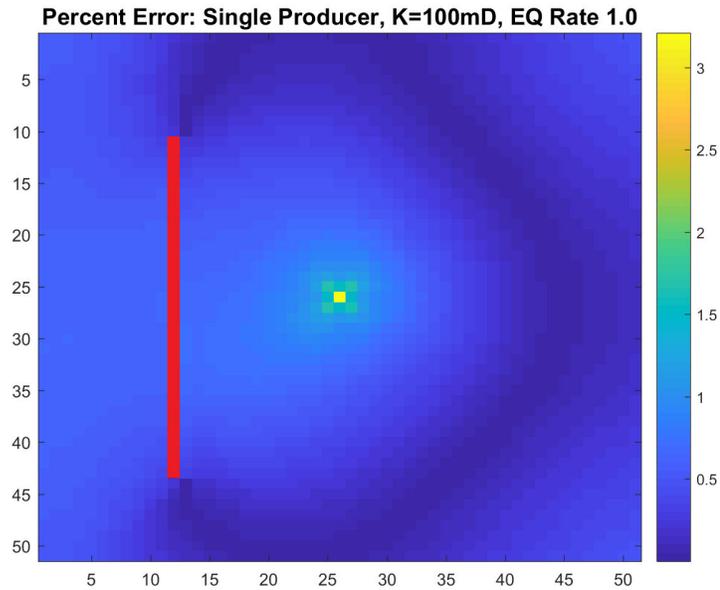


Figure 3.27 : Cell by cell percent error of the single barrier, 100 mD test.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
1690	2.46	0.05%

Table 3.15 : The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, single producer test. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

of the well.

Double Producer, with Central Barrier and Varied Permeability

Figure 3.35 shows the domain of this test. Varied permeability values of 10 mD, 100 mD, and 1000 mD were used in this simulation, again shown in figure 3.35. The remaining variables are listed in tables 3.1, 3.2, and 3.3. As in the previous

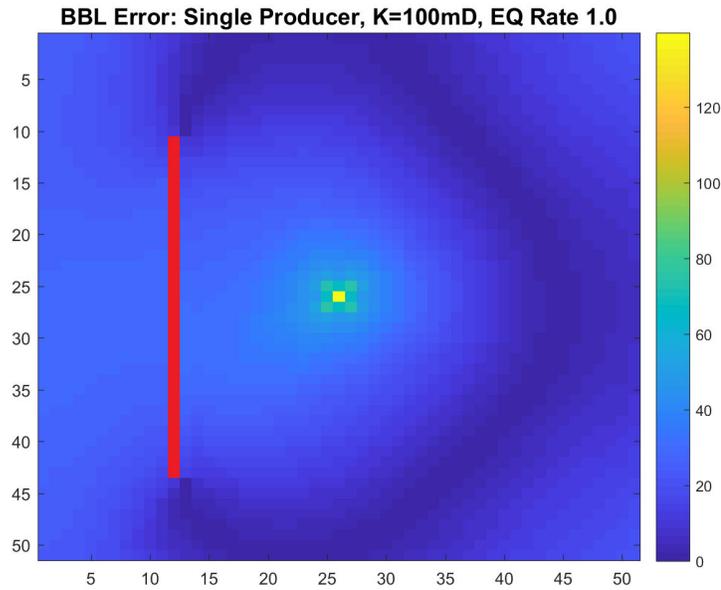


Figure 3.28 : Cell by cell barrels of oil error of the single barrier, 100 mD test.

<u>Norm</u>	<u>Percent Error</u>
∞	4.54%
2	0.75%
1	0.44%

Table 3.16 : Percent errors for the double barrier, double producer, varied permeability test, with a left barrier.

impermeable barrier tests, a high relative error is observed around the barriers, in every EQ rate comparison. Therefore, minimizing the error along the barrier (hence, the ℓ_∞ norm) is determined to be the metric for this reservoir test.

This test setup corresponded to a best-match equalization rate of 0.4; see Table A.25 for additional details. Table 3.18 shows the ℓ_∞ , ℓ_2 , and ℓ_1 norm errors in

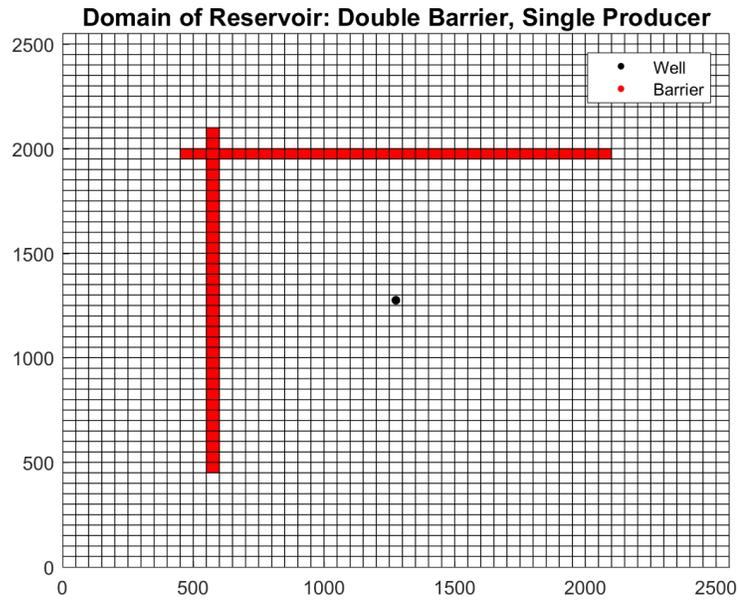


Figure 3.29 : The domain for the double barrier, single producer test. Red cells represent the location of the barrier.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
9333	13.60	0.18%

Table 3.17 : The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, double producer, varied permeability test, with a left barrier. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

percentage form. Further, figure 3.36 shows the error heat map in terms of percent error, and figure 3.37 shows the error heat map in terms of barrels of oil. Again, note that the areas of highest error in the domain are near the production wells, as is to be expected. Furthermore, the ℓ_∞ error in this test is approximately 1.5% less than

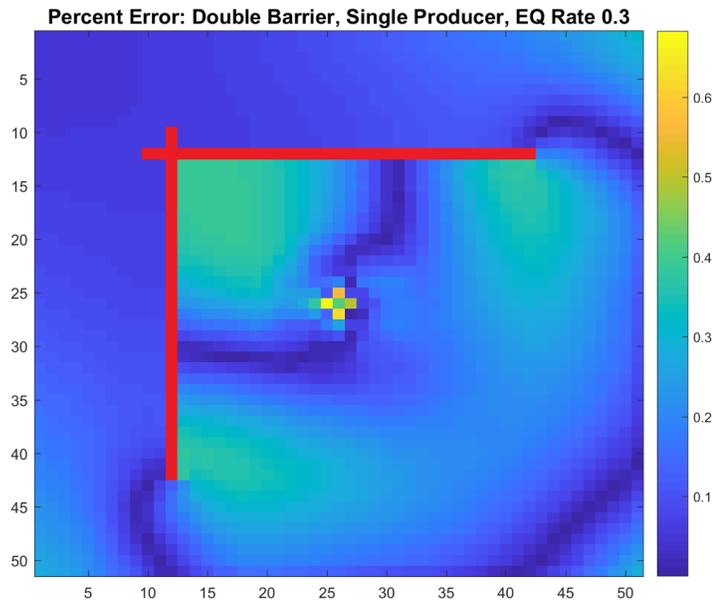


Figure 3.30 : Cell by cell percent error of the double barrier, single producer test.

the ℓ_∞ error in the central barrier test.

<u>Norm</u>	<u>Percent Error</u>
∞	2.97%
2	0.50%
1	0.37%

Table 3.18 : Percent errors for the double barrier, double producer, varied permeability test, with a central barrier.

Finally, the error between the amount of oil remaining in a 40 acre area around the well head in the two models is considered, for an EQ rate of 0.4. Table 3.19 shows that the two models differ by 0.12% in terms of total oil remaining, within 40 acres

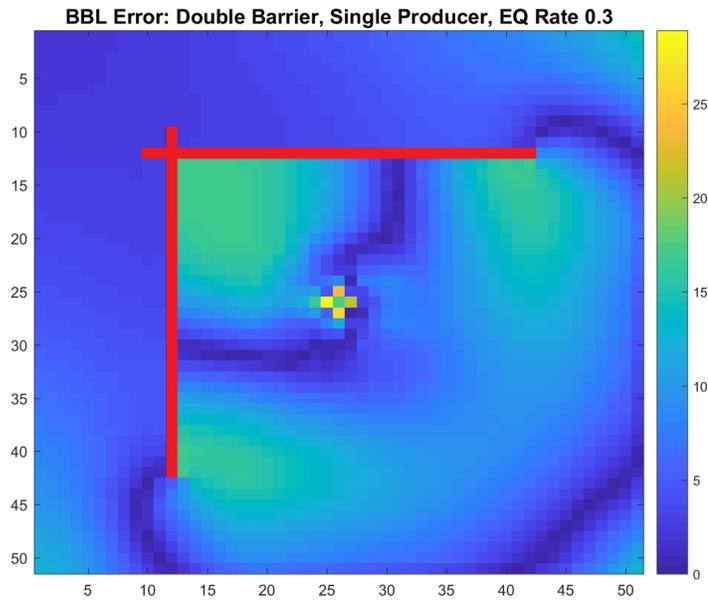


Figure 3.31 : Cell by cell barrels of oil error of the double barrier, single producer test.

of the well.

<u>Difference (bbl)</u>	<u>Average Difference (bbl/cell)</u>	<u>Difference (%)</u>
6225	9.07	0.12%

Table 3.19 : The 40 acre difference between the traditional and ReservoirGrail simulators, for the double barrier, double producer, varied permeability test, with a central barrier. A positive difference implies that the traditional simulator reports more remaining oil than ReservoirGrail does.

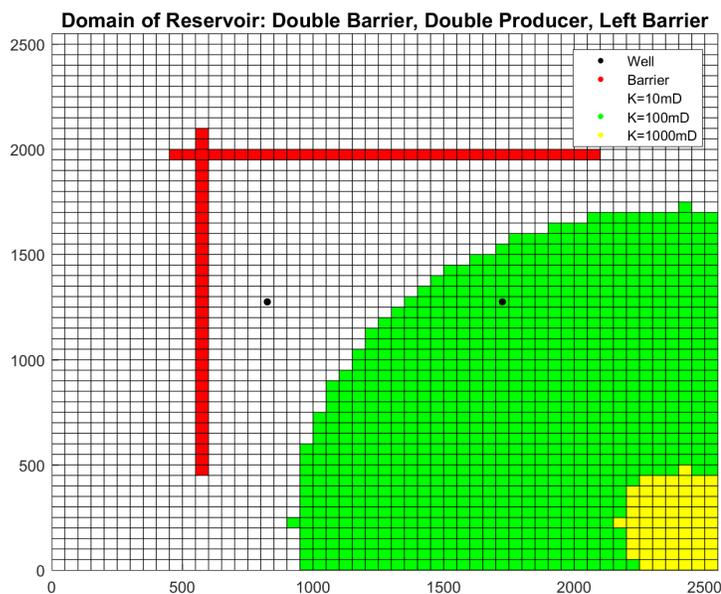


Figure 3.32 : The domain for the double barrier, double producer test with varied permeability and a barrier to the left of both production wells.

3.2.5 Small Permeability and EQ Rate Tests

As depicted in figures 3.3, 3.5, and 3.6, and discussed briefly in Section 3.2.1, a low permeability value of 10mD leads to a higher relative error when compared to the error generated by reservoirs with larger permeability values. This section will show that this high relative error is due to the coarse EQ rate steps taken in previous tests, and not due to the divergence of the two models as permeability and EQ rate decrease.

I first consider a reservoir with permeability of 10mD, and EQ rates of 0.00, 0.04, 0.07, 0.10, 0.13, and 0.16, with a domain given in figure 3.1. This is done in order to compare results from this test to results generated in Section 3.2.1, where a 10mD reservoir was also tested, but with a coarser EQ rate step of 0.1. I will show error in

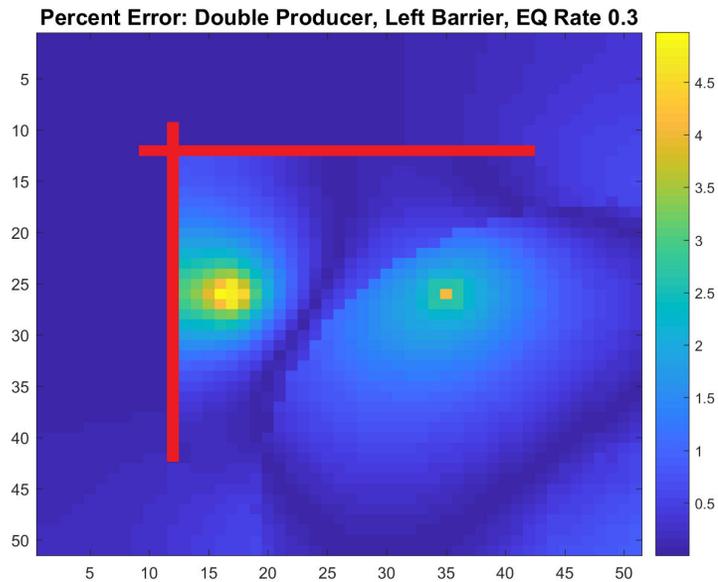


Figure 3.33 : Cell by cell percent error of the double barrier, double producer, varied permeability test, with a left barrier.

the ℓ_∞ norm, in order to more effectively demonstrate the difference that a smaller EQ rate step makes; however, Table A.27 shows results for the ℓ_2 and ℓ_1 errors as well, which depict the same behavior.

Figure 3.38 shows the results of this test. Furthermore, figure 3.39 shows how the ℓ_∞ error changes based on the step size of EQ rate. The data in this figure is cutoff after the 40mD test, in order to provide a closer look at this range of K values.

If an EQ rate step of 0.1 is taken as in Section 3.2.1, the best match EQ rate for this domain is 0.1, which leads to an ℓ_∞ error of approximately 4.78%. This is much higher than the errors generated for the 20, 30, and 40mD tests, all of which are below 1% ℓ_∞ error. However, if a smaller EQ rate step is taken, the best match EQ rate is 0.04, which corresponds to an ℓ_∞ error of 2.43%; as this error is much closer

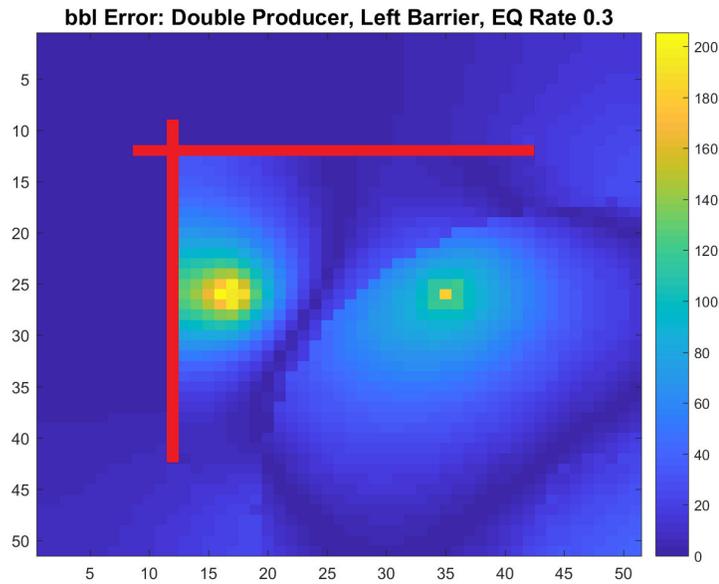


Figure 3.34 : Cell by cell barrels of oil error of the double barrier, double producer, varied permeability test, with a left barrier.

to the errors generated by the 20, 30, and 40mD tests in Section 3.2.1, this implies that the high relative error depicted in figure 3.6 is due to a coarse EQ rate step, and not due to any kind of divergence between the two simulators.

I also ran this test with permeability values of 8mD and 12mD, in order to show that the same results hold if the permeability is slightly modified. I again use the ℓ_∞ error to show my result with more clarity, but tables A.26 and A.28 shows results for the ℓ_2 and ℓ_1 errors as well, where the same behavior holds. Figure 3.40 shows the EQ rate vs. ℓ_∞ error generated for an 8mD reservoir, and analogously, figure ?? shows the EQ rate vs. ℓ_∞ error generated for a 12mD reservoir. In both cases, note that a lower error is attainable when smaller EQ rate steps are taken, as shown in the 10mD test.

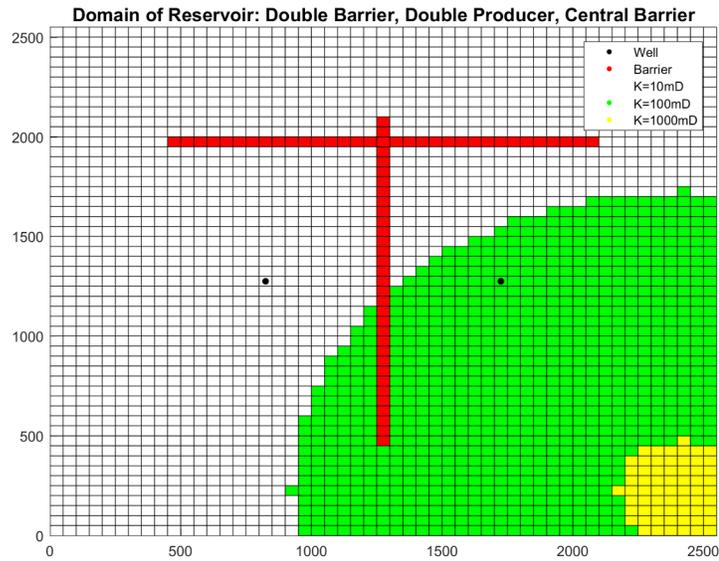


Figure 3.35 : The domain for the double barrier, double producer test with varied permeability and a barrier in between both production wells.

3.2.6 Refined Mesh Tests

All of the previous tests have been performed on a mesh that is 51×51 , as depicted in figures 3.1 and 3.2. To determine the effect that a more refined mesh has on the resulting difference between the traditional model and ReservoirGrail, results are generated using the traditional model and a mesh that is 102×102 with half the spacing between cells. Once these results are obtained, the generated 102×102 matrix containing the mass in each element is converted into a 51×51 matrix, in order to compare results with ReservoirGrail. To perform this conversion, the 102×102 matrix is broken up into distinct 2×2 sub-matrices, each of whose entry-wise sum corresponds to the mass in an element in the 51×51 mesh. As an example, the mass of the elements in the 102×102 mesh with coordinates $(0, 25) \times (0, 25)$,

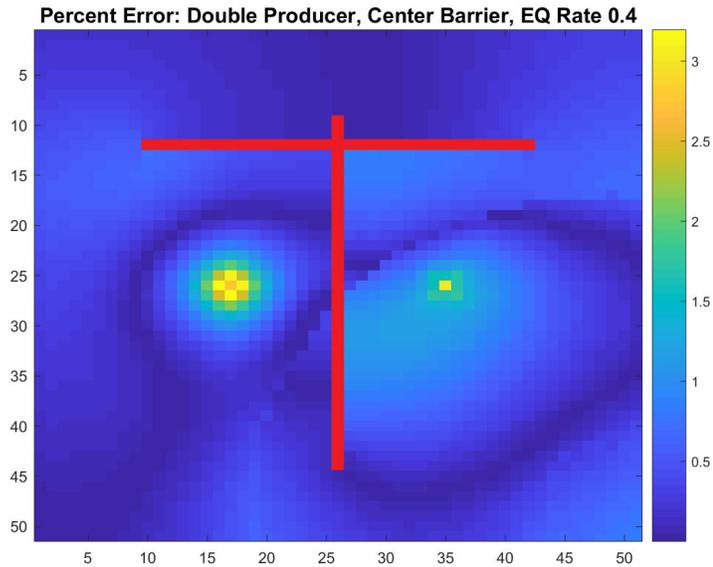


Figure 3.36 : Cell by cell percent error of the double barrier, double producer, varied permeability test, with a central barrier.

$(0, 25) \times (25, 50)$, $(25, 50) \times (0, 25)$, and $(25, 50) \times (25, 50)$ are summed to obtain the mass in the $(0, 50) \times (0, 50)$ section of the domain, which is then compared to the mass reported by ReservoirGrail in the same $(0, 50) \times (0, 50)$ section of the domain.

Two reservoirs are considered with refined meshes: a domain with a single production well and a constant permeability of 30mD, analogous to tests performed in Section 3.2.1, and a domain with a single production well, constant permeability of 30mD, and an impermeable barrier to the left of the well, analogous to tests performed in Section 3.2.3. Figure 3.42 shows the domain used in this first test.

As in Section 3.2.1, the ℓ_2 error was deemed to be the most important error metric in this test, so the best-match EQ rate is defined in this case to be the EQ rate which minimizes the ℓ_2 difference between the two models. This test setup corresponded to

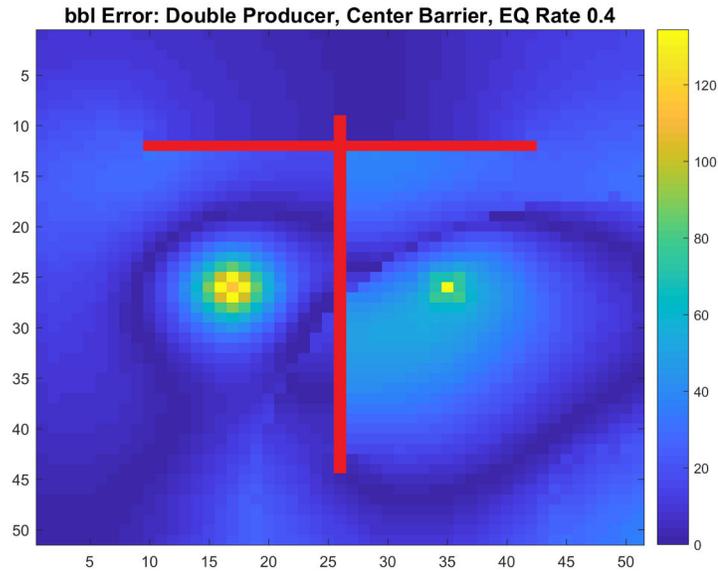


Figure 3.37 : Cell by cell barrels of oil error of the double barrier, double producer, varied permeability test, with a central barrier.

a best-match EQ rate of 0.7, which matches the best-match EQ rate obtained when simulating with the coarser 51×51 mesh. Figure 3.43 shows the EQ rate vs. $\%l_2$ error, for the test with the coarse mesh, as well as the test with the finer mesh.

Table 3.20 shows the l_∞ , l_2 , and l_1 norm errors in percentage form. Note that although simulating with a finer mesh reduces the l_∞ difference slightly between the two models, there is no other significant change between the two sets of results.

Figure 3.44 shows the domain used in the second test with the refined mesh, which contains an impermeable barrier to the left of the well, analogous to tests performed in Section 3.2.3.

As in Section 3.2.3, the l_∞ error was deemed to be the most important error metric in this test, so the best-match EQ rate is defined in this case to be the EQ rate which

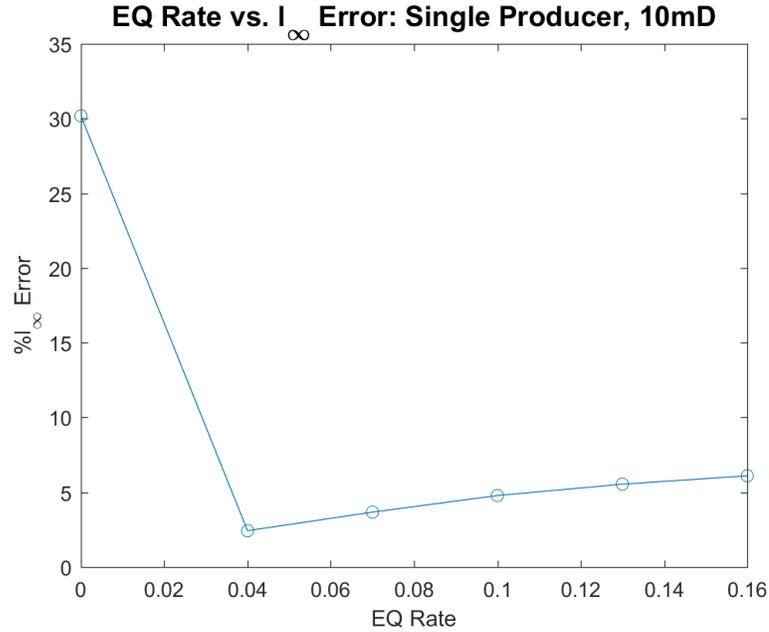


Figure 3.38 : EQ rate vs. l_∞ error of the 10mD permeability test.

<u>Norm</u>	<u>Percent Error, 51×51 Mesh</u>	<u>Percent Error, 102×102 Mesh</u>
∞	0.96%	0.92%
2	0.10%	0.10%
1	0.08%	0.08%

Table 3.20 : Percent errors for the constant permeability of 30mD test, with both a coarse and fine mesh.

minimizes the l_∞ difference between the two models. This test setup corresponded to a best-match EQ rate of 0.4, which matches the best-match EQ rate obtained when simulating with the coarser 51×51 mesh. Figure 3.45 shows the EQ rate vs. $\%l_\infty$ error, for the test with the coarse mesh, as well as the test with the finer mesh.

Table 3.21 shows the l_∞ , l_2 , and l_1 norm errors in percentage form. Although the

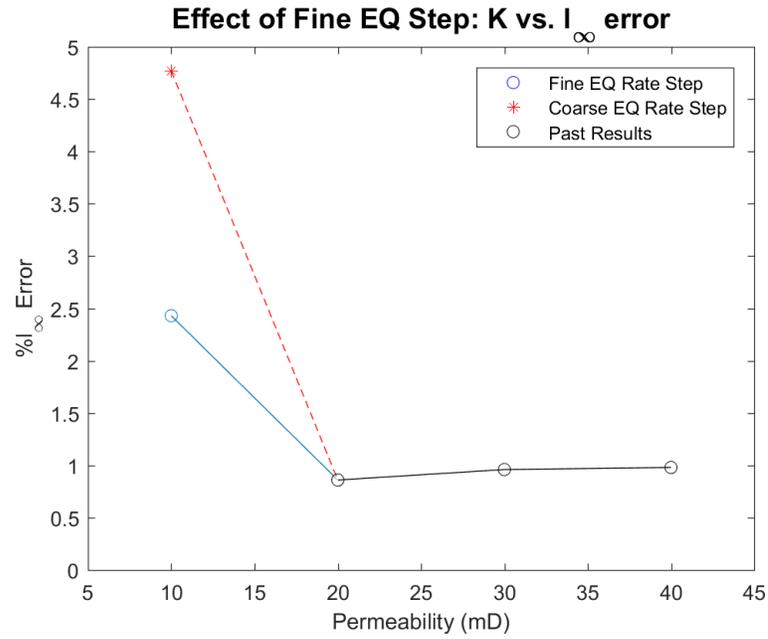


Figure 3.39 : Comparison between l_∞ error when a coarse EQ rate step is used vs. a finer step.

finer mesh gives a slightly higher l_∞ difference with this setup, the change is small.

Norm	Percent Error, 51×51 Mesh	Percent Error, 102×102 Mesh
∞	0.66%	0.69%
2	0.15%	0.15%
1	0.13%	0.13%

Table 3.21 : Percent errors for the single fault test, with both a coarse and fine mesh.

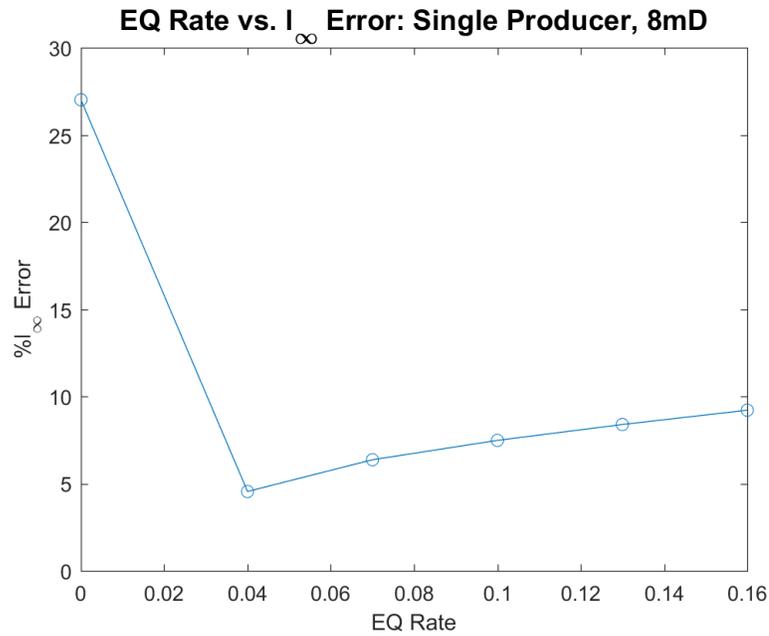


Figure 3.40 : EQ rate vs. l_∞ error of the 8mD permeability test.

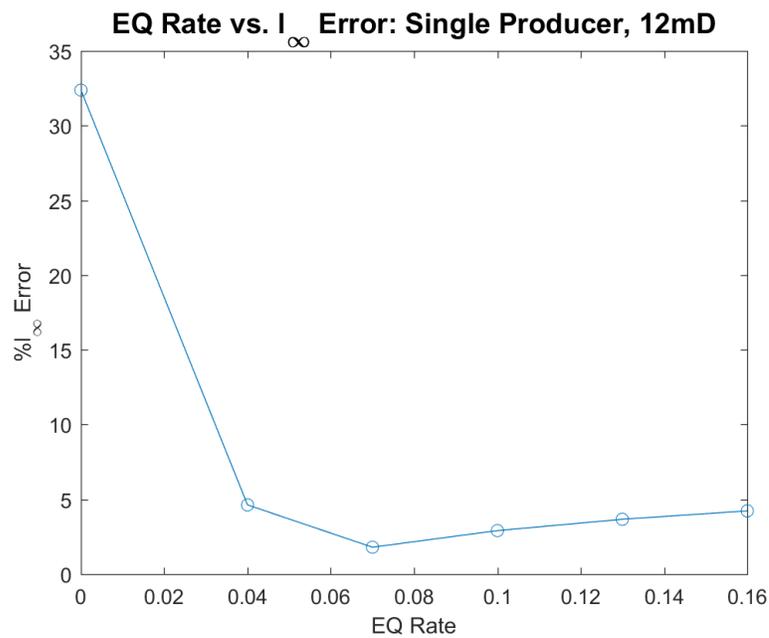


Figure 3.41 : EQ rate vs. l_∞ error of the 12mD permeability test.

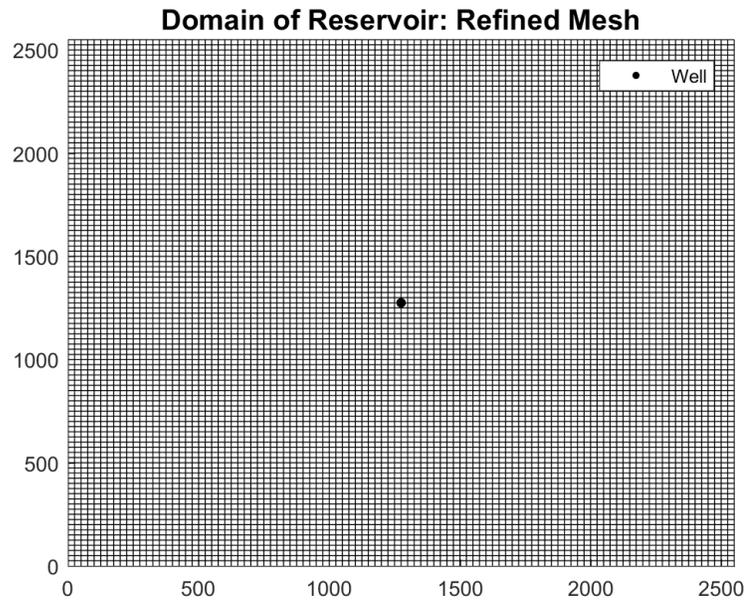


Figure 3.42 : Domain for the refined mesh test, with constant 30mD permeability.

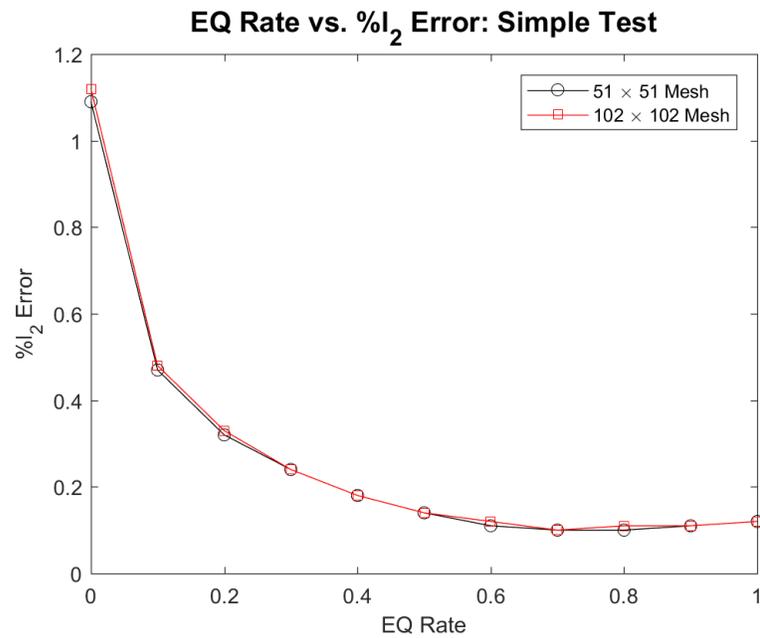


Figure 3.43 : Relationship between EQ rate and %l₂ difference between the simulators, for the coarse and fine mesh; see table A.29 for more details.

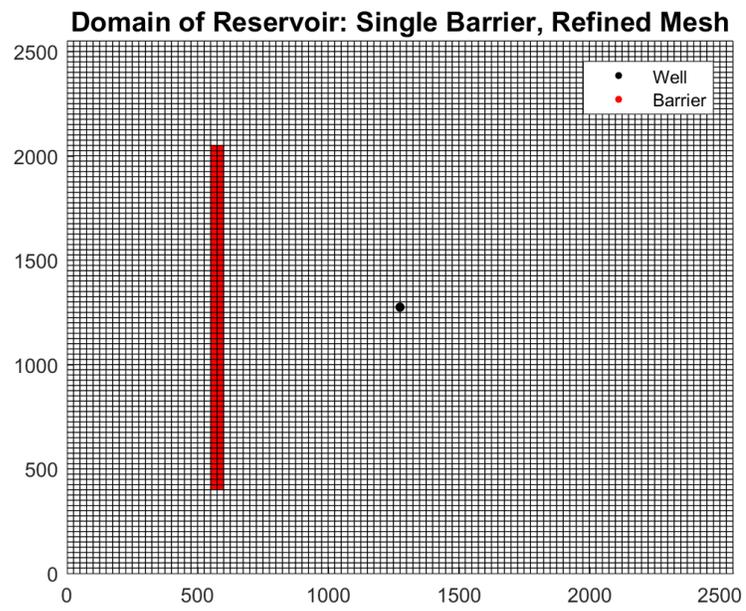


Figure 3.44 : Domain for the refined mesh test, with constant 30mD permeability and a single barrier.

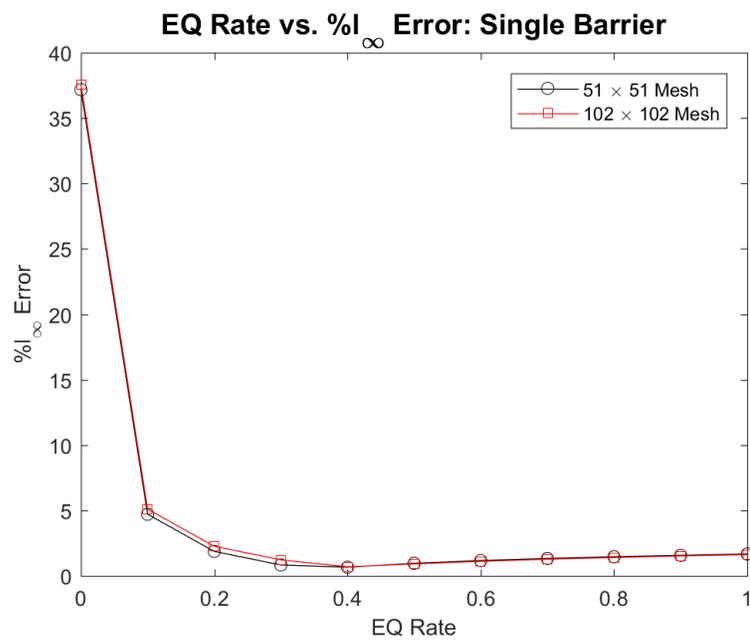


Figure 3.45 : Relationship between EQ rate and $\%l_2$ difference between the simulators, for the coarse and fine mesh with a single barrier; see table A.30 for more details.

Chapter 4

Future Work

4.1 Future Work

I have shown that a specific agent-based model (ReservoirGrail) when applied to simple oil reservoir scenarios gives results that differ by no more than 0.6% than the results obtained through a traditional model, when error is measured in ℓ_2 . Similarly low error is obtained when measuring error in the ℓ_1 and ℓ_∞ norms, with differences of no more than 0.4% in ℓ_1 , and no more than 5.4% in ℓ_∞ . Similar results are obtained when the simulators are tested with domains with varied permeability and impermeable barriers: ℓ_2 errors of 1.21% for the varied permeability domain and 0.75% for the impermeable barrier domains are observed. In addition to this, I have also shown that there is a correlation between ReservoirGrail's artificial parameter, equalization rate, and the physical ratio present in traditional models, K/μ , and have further demonstrated that this relationship is more sensitive when considering reservoirs with permeability values less than 10mD.

From this point, there are multiple avenues for future directions of research in this project. Obtaining a more precise equalization rate to K/μ ratio correlation would allow for a more precise correspondence between physical reservoir parameter's and ReservoirGrail's EQ rate. Furthermore, introducing 2 and 3 phase flow comparisons to ReservoirGrail would allow for more realistic scenarios, and would either provide more confidence in ReservoirGrail's results or suggest areas of improvement. In addition,

comparing to ReservoirGrail's results to those of a 2 phase flow simulator would allow for the scenario of multiple production and injection wells running in the same reservoir, which is a very common practice in industry.

Appendices

Appendix A

Tables for Reservoir Tests

This appendix has the raw data used to determine the figures in Sections 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, and 3.2.6.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.39%	0.80%	30.14%
0.1	0.37%	0.55%	4.77%
0.2	0.42%	0.70%	6.77%
0.3	0.46%	0.80%	7.82%
0.4	0.50%	0.88%	8.39%
0.5	0.54%	0.93%	8.74%
0.6	0.57%	0.98%	9.00%
0.7	0.60%	1.01%	9.20%
0.8	0.62%	1.05%	9.35%
0.9	0.64%	1.07%	9.48%
1.0	0.66%	1.09%	9.58%

Table A.1 : Error comparisons between different EQ rates, for the single producer scenario with permeability 10mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.15%	0.92%	36.97%
0.1	0.08%	0.21%	2.85%
0.2	0.05%	0.08%	0.86%
0.3	0.06%	0.11%	1.74%
0.4	0.09%	0.17%	2.25%
0.5	0.12%	0.22%	2.59%
0.6	0.14%	0.26%	2.83%
0.7	0.17%	0.30%	3.01%
0.8	0.19%	0.33%	3.16%
0.9	0.21%	0.35%	3.28%
1.0	0.22%	0.38%	3.37%

Table A.2 : Error comparisons between different EQ rates, for the single producer scenario with permeability 20mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.28%	1.09%	39.43%
0.1	0.24%	0.47%	5.31%
0.2	0.21%	0.32%	2.51%
0.3	0.17%	0.24%	1.46%
0.4	0.13%	0.18%	0.90%
0.5	0.11%	0.14%	0.54%
0.6	0.09%	0.11%	0.78%
0.7	0.08%	0.103*%	0.96%
0.8	0.07%	0.104*%	1.11%
0.9	0.07%	0.11%	1.22%
1.0	0.08%	0.12%	1.32%

Table A.3 : Error comparisons between different EQ rates, for the single producer scenario with permeability 30mD.

*: Three decimal places were given here in order to show which EQ rate minimized the given norm; with only two decimal places, this determination would be non-unique.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.38%	1.20%	40.70%
0.1	0.34%	0.62%	6.58%
0.2	0.30%	0.47%	3.78%
0.3	0.27%	0.38%	2.73%
0.4	0.23%	0.31%	2.17%
0.5	0.20%	0.26%	1.81%
0.6	0.17%	0.22%	1.55%
0.7	0.15%	0.19%	1.36%
0.8	0.13%	0.16%	1.20%
0.9	0.11%	0.14%	1.08%
1.0	0.10%	0.12%	0.98%

Table A.4 : Error comparisons between different EQ rates, for the single producer scenario with permeability 40mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.44%	1.27%	41.48%
0.1	0.40%	0.71%	7.36%
0.2	0.36%	0.56%	4.57%
0.3	0.33%	0.47%	3.51%
0.4	0.29%	0.40%	2.95%
0.5	0.26%	0.35%	2.59%
0.6	0.23%	0.30%	2.34%
0.7	0.20%	0.27%	2.14%
0.8	0.18%	0.24%	1.98%
0.9	0.16%	0.21%	1.86%
1.0	0.15%	0.19%	1.76%

Table A.5 : Error comparisons between different EQ rates, for the single producer scenario with permeability 50mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.56%	1.41%	43.13%
0.1	0.53%	0.89%	9.01%
0.2	0.49%	0.74%	6.22%
0.3	0.45%	0.65%	5.17%
0.4	0.41%	0.58%	4.60%
0.5	0.38%	0.52%	4.25%
0.6	0.35%	0.48%	3.99%
0.7	0.32%	0.44%	3.79%
0.8	0.30%	0.41%	3.64%
0.9	0.28%	0.38%	3.51%
1.0	0.27%	0.36%	3.41%

Table A.6 : Error comparisons between different EQ rates, for the single producer scenario with permeability 100mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.62%	1.49%	44.07%
0.1	0.59%	0.98%	9.95%
0.2	0.55%	0.83%	7.16%
0.3	0.51%	0.74%	6.11%
0.4	0.47%	0.67%	5.54%
0.5	0.44%	0.61%	5.19%
0.6	0.41%	0.57%	4.93%
0.7	0.38%	0.53%	4.73%
0.8	0.36%	0.50%	4.58%
0.9	0.34%	0.47%	4.45%
1.0	0.33%	0.45%	4.35%

Table A.7 : Error comparisons between different EQ rates, for the single producer scenario with permeability 200mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.66%	1.54%	44.76%
0.1	0.62%	1.03%	10.63%
0.2	0.59%	0.89%	7.84%
0.3	0.55%	0.79%	6.79%
0.4	0.51%	0.72%	6.23%
0.5	0.48%	0.67%	5.87%
0.6	0.45%	0.62%	5.61%
0.7	0.42%	0.59%	5.42%
0.8	0.40%	0.55%	5.26%
0.9	0.38%	0.53%	5.13%
1.0	0.36%	0.50%	5.03%

Table A.8 : Error comparisons between different EQ rates, for the single producer scenario with permeability 500mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.67%	1.56%	45.04%
0.1	0.64%	1.05%	10.92%
0.2	0.60%	0.91%	8.13%
0.3	0.56%	0.81%	7.07%
0.4	0.52%	0.74%	6.51%
0.5	0.49%	0.69%	6.15%
0.6	0.46%	0.64%	5.90%
0.7	0.43%	0.60%	5.70%
0.8	0.41%	0.57%	5.54%
0.9	0.39%	0.54%	5.42%
1.0	0.37%	0.52%	5.32%

Table A.9 : Error comparisons between different EQ rates, for the single producer scenario with permeability 1000mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.29%	0.59%	16.30%
0.1	0.26%	0.33%	1.99%
0.2	0.30%	0.44%	2.94%
0.3	0.34%	0.51%	3.40%
0.4	0.38%	0.56%	3.68%
0.5	0.41%	0.60%	3.86%
0.6	0.44%	0.64%	3.99%
0.7	0.46%	0.66%	4.09%
0.8	0.48%	0.69%	4.17%
0.9	0.50%	0.71%	4.23%
1.0	0.51%	0.72%	4.28%

Table A.10 : Error comparisons between different EQ rates, for the double producer scenario with permeability 10mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.18%	0.71%	19.53%
0.1	0.13%	0.21%	1.67%
0.2	0.10%	0.12%	0.32%
0.3	0.08%	0.10%	0.69%
0.4	0.08%	0.11%	0.94%
0.5	0.09%	0.14%	1.11%
0.6	0.10%	0.16%	1.23%
0.7	0.12%	0.18%	1.33%
0.8	0.14%	0.20%	1.40%
0.9	0.15%	0.22%	1.46%
1.0	0.16%	0.23%	1.50%

Table A.11 : Error comparisons between different EQ rates, for the double producer scenario with permeability 20mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.31%	0.84%	20.75%
0.1	0.26%	0.39%	2.85%
0.2	0.22%	0.29%	1.46%
0.3	0.18%	0.23%	0.93%
0.4	0.15%	0.18%	0.66%
0.5	0.12%	0.15%	0.48%
0.6	0.10%	0.12%	0.35%
0.7	0.08%	0.11%	0.36%
0.8	0.07%	0.093*%	0.43%
0.9	0.07%	0.087*%	0.49%
1.0	0.07%	0.086*%	0.54%

Table A.12 : Error comparisons between different EQ rates, for the double producer scenario with permeability 30mD.

*: Three decimal places were given here in order to show which EQ rate minimized the given norm; with only two decimal places, this determination would be non-unique.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.38%	0.92%	21.39%
0.1	0.34%	0.50%	3.47%
0.2	0.30%	0.39%	2.07%
0.3	0.26%	0.33%	1.55%
0.4	0.22%	0.28%	1.27%
0.5	0.19%	0.24%	1.10%
0.6	0.17%	0.21%	0.97%
0.7	0.15%	0.18%	0.87%
0.8	0.13%	0.16%	0.80%
0.9	0.11%	0.14%	0.73%
1.0	0.10%	0.13%	0.68%

Table A.13 : Error comparisons between different EQ rates, for the double producer scenario with permeability 40mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.43%	0.97%	21.78%
0.1	0.39%	0.56%	3.85%
0.2	0.35%	0.46%	2.46%
0.3	0.31%	0.39%	1.93%
0.4	0.27%	0.34%	1.66%
0.5	0.24%	0.30%	1.48%
0.6	0.22%	0.27%	1.35%
0.7	0.19%	0.24%	1.26%
0.8	0.17%	0.22%	1.18%
0.9	0.16%	0.20%	1.12%
1.0	0.14%	0.18%	1.06%

Table A.14 : Error comparisons between different EQ rates, for the double producer scenario with permeability 50mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.53%	1.07%	22.63%
0.1	0.49%	0.69%	4.67%
0.2	0.45%	0.59%	3.28%
0.3	0.41%	0.52%	2.75%
0.4	0.37%	0.47%	2.47%
0.5	0.34%	0.43%	2.29%
0.6	0.31%	0.39%	2.17%
0.7	0.29%	0.37%	2.07%
0.8	0.27%	0.34%	1.99%
0.9	0.25%	0.32%	1.93%
1.0	0.24%	0.30%	1.88%

Table A.15 : Error comparisons between different EQ rates, for the double producer scenario with permeability 100mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.58%	1.13%	23.11%
0.1	0.54%	0.76%	5.14%
0.2	0.50%	0.65%	3.74%
0.3	0.46%	0.59%	3.22%
0.4	0.42%	0.53%	2.94%
0.5	0.39%	0.49%	2.76%
0.6	0.36%	0.46%	2.64%
0.7	0.34%	0.43%	2.54%
0.8	0.32%	0.41%	2.46%
0.9	0.30%	0.38%	2.40%
1.0	0.29%	0.37%	2.35%

Table A.16 : Error comparisons between different EQ rates, for the double producer scenario with permeability 200mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.61%	1.16%	23.46%
0.1	0.57%	0.79%	5.48%
0.2	0.53%	0.69%	4.09%
0.3	0.49%	0.63%	3.56%
0.4	0.45%	0.57%	3.28%
0.5	0.42%	0.53%	3.10%
0.6	0.39%	0.50%	2.98%
0.7	0.37%	0.47%	2.88%
0.8	0.35%	0.44%	2.80%
0.9	0.33%	0.42%	2.74%
1.0	0.32%	0.41%	2.69%

Table A.17 : Error comparisons between different EQ rates, for the double producer scenario with permeability 500mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.62%	1.18%	23.61%
0.1	0.58%	0.81%	5.63%
0.2	0.54%	0.71%	4.23%
0.3	0.50%	0.64%	3.70%
0.4	0.46%	0.59%	3.42%
0.5	0.43%	0.55%	3.25%
0.6	0.40%	0.51%	3.12%
0.7	0.38%	0.48%	3.02%
0.8	0.36%	0.46%	2.95%
0.9	0.34%	0.44%	2.88%
1.0	0.33%	0.42%	2.83%

Table A.18 : Error comparisons between different EQ rates, for the double producer scenario with permeability 1000mD.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	1.33%	1.87%	35.93%
0.1	0.94%	1.21%	3.52%
0.2	0.84%	1.10%	5.60%
0.3	0.77%	1.05%	6.55%
0.4	0.73%	1.033*%	7.11%
0.5	0.71%	1.03%	7.50%
0.6	0.695*%	1.03%	7.78%
0.7	0.692*%	1.04%	8.02%
0.8	0.694*%	1.05%	8.21%
0.9	0.70%	1.06%	8.37%
1.0	0.70%	1.08%	8.50%

Table A.19 : Error comparisons between different EQ rates, for the varied permeability test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.27%	1.01%	37.17%
0.1	0.20%	0.37%	4.71%
0.2	0.17%	0.23%	1.87%
0.3	0.14%	0.17%	0.83%
0.4	0.13%	0.15%	0.66%
0.5	0.12%	0.14%	0.96%
0.6	0.12%	0.15%	1.17%
0.7	0.12%	0.16%	1.33%
0.8	0.12%	0.17%	1.46%
0.9	0.13%	0.19%	1.57%
1.0	0.14%	0.21%	1.67%

Table A.20 : Error comparisons between different EQ rates, for the single barrier, 30mD test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.42%	1.19%	39.39%
0.1	0.39%	0.63%	6.85%
0.2	0.36%	0.50%	4.01%
0.3	0.33%	0.43%	2.97%
0.4	0.30%	0.38%	2.44%
0.5	0.28%	0.34%	2.12%
0.6	0.26%	0.31%	1.89%
0.7	0.24%	0.28%	1.72%
0.8	0.22%	0.26%	1.58%
0.9	0.20%	0.24%	1.46%
1.0	0.19%	0.22%	1.35%

Table A.21 : Error comparisons between different EQ rates, for the single barrier, 50mD test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.60%	1.37%	41.28%
0.1	0.57%	0.86%	8.64%
0.2	0.54%	0.74%	5.78%
0.3	0.51%	0.67%	4.74%
0.4	0.48%	0.62%	4.21%
0.5	0.46%	0.58%	3.89%
0.6	0.44%	0.55%	3.67%
0.7	0.41%	0.52%	3.49%
0.8	0.39%	0.49%	3.35%
0.9	0.38%	0.47%	3.23%
1.0	0.36%	0.44%	3.12%

Table A.22 : Error comparisons between different EQ rates, for the single barrier, 100mD test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.29%	0.98%	36.21%
0.1	0.21%	0.31%	4.09%
0.2	0.17%	0.21%	1.35%
0.3	0.16%	0.19%	0.64%
0.4	0.16%	0.20%	1.05%
0.5	0.17%	0.21%	1.31%
0.6	0.18%	0.23%	1.48%
0.7	0.18%	0.24%	1.62%
0.8	0.19%	0.26%	1.73%
0.9	0.19%	0.27%	1.82%
1.0	0.20%	0.29%	1.91%

Table A.23 : Error comparisons between different EQ rates, for the double barrier, single producer, 30mD test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.66%	1.28%	29.32%
0.1	0.52%	0.86%	6.69%
0.2	0.47%	0.78%	4.73%
0.3	0.44%	0.75%	4.54%
0.4	0.42%	0.72%	4.74%
0.5	0.40%	0.71%	4.86%
0.6	0.39%	0.70%	4.96%
0.7	0.37%	0.69%	5.03%
0.8	0.36%	0.69%	5.09%
0.9	0.35%	0.69%	5.15%
1.0	0.35%	0.68%	5.20%

Table A.24 : Error comparisons between different EQ rates, for the double barrier, double producer, variable permeability with a left barrier test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.60%	1.21%	28.10%
0.1	0.44%	0.67%	5.84%
0.2	0.41%	0.58%	3.97%
0.3	0.39%	0.53%	3.29%
0.4	0.37%	0.50%	2.97%
0.5	0.35%	0.48%	3.16%
0.6	0.34%	0.47%	3.29%
0.7	0.33%	0.47%	3.39%
0.8	0.33%	0.46%	3.47%
0.9	0.32%	0.46%	3.54%
1.0	0.32%	0.46%	3.60%

Table A.25 : Error comparisons between different EQ rates, for the double barrier, double producer, variable permeability with a central barrier test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.00	0.53%	0.92%	27.01%
0.04	0.52%	0.73%	4.55%
0.07	0.53%	0.79%	6.37%
0.10	0.54%	0.85%	7.48%
0.13	0.55%	0.91%	8.39%
0.16	0.57%	0.95%	9.21%

Table A.26 : Error comparisons between different EQ rates, for the single barrier and 8mD permeability test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.00	0.39%	0.81%	30.18%
0.04	0.36%	0.47%	2.43%
0.07	0.37%	0.50%	3.67%
0.10	0.38%	0.56%	4.78%
0.13	0.39%	0.61%	5.54%
0.16	0.41%	0.65%	6.10%

Table A.27 : Error comparisons between different EQ rates, for the single barrier and 10mD permeability test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.00	0.29%	0.79%	32.38%
0.04	0.24%	0.31%	4.63%
0.07	0.24%	0.30%	1.80%
0.10	0.25%	0.34%	2.91%
0.13	0.27%	0.39%	3.67%
0.16	0.28%	0.43%	4.23%

Table A.28 : Error comparisons between different EQ rates, for the single barrier and 12mD permeability test.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.29%	1.12%	40.15%
0.1	0.25%	0.48%	5.74%
0.2	0.21%	0.33%	2.92%
0.3	0.17%	0.24%	1.86%
0.4	0.14%	0.18%	1.29%
0.5	0.11%	0.14%	0.93%
0.6	0.09%	0.12%	0.73%
0.7	0.08%	0.10%	0.92%
0.8	0.07%	0.11%	1.06%
0.9	0.07%	0.11%	1.18%
1.0	0.08%	0.12%	1.28%

Table A.29 : Error comparisons between different EQ rates, for the refined traditional model mesh with constant 30mD permeability.

<u>EQ Rate</u>	<u>1-Norm Error (%)</u>	<u>2-Norm Error (%)</u>	<u>∞-Norm Error (%)</u>
0.0	0.27%	1.02%	37.55%
0.1	0.21%	0.37%	5.10%
0.2	0.17%	0.23%	2.26%
0.3	0.15%	0.17%	1.22%
0.4	0.13%	0.15%	0.69%
0.5	0.12%	0.14%	0.91%
0.6	0.12%	0.15%	1.11%
0.7	0.12%	0.16%	1.27%
0.8	0.13%	0.17%	1.40%
0.9	0.13%	0.19%	1.52%
1.0	0.14%	0.21%	1.62%

Table A.30 : Error comparisons between different EQ rates, for the refined traditional model mesh with constant 30mD permeability and a single barrier.

Appendix B

Unit Descriptions and Conversions

In order to make this thesis as applicable to real scenarios as possible, I decided to use units that are common among the industry in the United States. However, most of these units are not SI units, or common in other countries. The purpose of this appendix then, is to give conversion factors between the units I use in this paper and SI units, as well as to mention realistic ranges of these units when applicable.

Compressibility: Denoted c , given in units of psi^{-1} (“psi” stands for “pounds per square inch”). The approximate range of values c can take for oil is between 10^{-3}psi^{-1} and 10^{-7}psi^{-1} . The conversion to SI units is given as

$$\frac{1}{\text{psi}} = 1.4503774 \times 10^{-4} \frac{\text{meter}^2}{\text{Newton}}$$

To justify the compressibility value I used for oil in Section 3.1.2, it will be necessary to describe how the compressibility of oil changes as a function of pressure. Oil compressibility decreases with pressure as shown in figure B.1, until it reaches the “bubble point” pressure, which is the amount of pressure required to dissolve the gas into the liquid oil. After this point, the compressibility changes at a much slower rate. As this paper assumes single phase flow (that is, no gas present), the compressibility of oil was taken to be $6.953932712 \times 10^{-6}$, which correlates to values in figure B.1 after the bubble point pressure.

Porosity: Denoted ϕ , this is a unitless quantity representing the ratio between the volume that is free for fluid to occupy to the total volume, in a given section of the

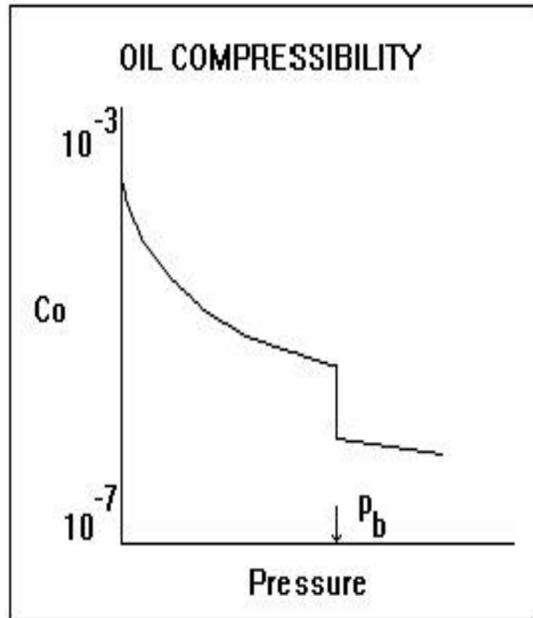


Figure B.1 : Oil compressibility (y-axis), as a function of pressure (x-axis) [16].

reservoir. Realistic values for ϕ range between 0.1 and 0.3. I chose a value of 1 for my tests, due to the limited initial functionality of the standard Darcy simulator, but I do not expect that changing this value would influence the results or error much at all, since it simply affects the speed at which the oil moves through the reservoir.

Viscosity: Denoted μ , and given in units of cP (centi-Poise). The approximate range of realistic values for the viscosity of oil is between 1 cP and 1000 cP, depending greatly on the temperature and pressure. The conversion to SI units is given as

$$1 \text{ cP} = 10^{-3} \text{ Pascal seconds}$$

A viscosity of $\mu = 10\text{cP}$ was used throughout all of the tests, which is a common value observed in practice.

Permeability: Denoted K , and given in units of mD (milli-Darcy's). The approximate range of values for the permeability of rocks found in oil reservoirs is between 10 mD and 1000 mD. The conversion to SI unites is given as

$$1 \text{ mD} = 9.901 \times 10^{-16} \text{ meter}^2$$

The permeability values were different in most of the tests that I performed, and took on values everywhere between 10 mD and 1000 mD. This wide variation was necessary in order to determine a relationship between the permeability in the traditional simulator and the EQ rate in ReservoirGrail.

Volume of oil: The volume of oil is given in units of bbl (barrels of oil). The conversion rate is given as

$$1 \text{ bbl} = 0.158983 \text{ meter}^3$$

Appendix C

ReservoirGrail: Additional Information

The purpose of this appendix is to provide some additional information pertaining to the inputs of ReservoirGrail, and how its algorithm works. Section C.1 describes how to format and import data for the production and injection wells in the domain, Section C.2 defines the reservoir map inputs used in the tests ran in this thesis, Section C.3 describes the less-intuitive global parameters in ReservoirGrail, and Section C.4 gives two flowcharts depicting how the well and production phases work, which are the two main components of ReservoirGrail’s algorithm.

C.1 Importing Data

In the middle of creating a new project, the dialogue box shown in figure C.1 will ask the user if they want to import production and injection data for the project now, or wait until later. I have personally found the process easier if I imported data at this point, compared to waiting until after the project has been created. Clicking “Yes” will bring up figure C.2, which asks the user to specify a header file for the project. The header file contains the API of each well in the domain, as well as their X and Y coordinates; figure C.3 shows an example of the header file I used for a single, centered production well in my tests. Checking the “CSV as a header row” option prior to importing the data lets ReservoirGrail automatically fill the “Data Type” options, corresponding to the header row. After importing the header file,

production and injection data can be specified for each well by importing the relevant CSV file. Figure C.4 shows how to import the production data, and figure C.5 shows an example of the production data I used for the tests in this thesis. As I limited my tests to single phase flow, I did not have any injection wells, and therefore no injection data to import.

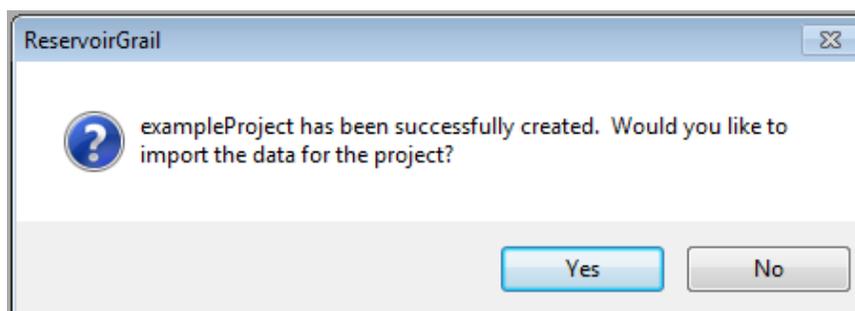


Figure C.1 : Dialogue box, prompting the user to import data for their project. It is easier to import this data now, rather than after the project has been created.

C.2 Defining Reservoir Maps

After creating the project and opening it, ReservoirGrail brings up its home project screen, shown in figure C.6. This screen asks for the type of system, pay thickness and flow path maps, and a description of the domain's discretization, in order to define the structure of the reservoir being simulated. The type of system being simulated is analogous to the boundary conditions imposed on the domain: a closed system corresponds to zero Neumann (i.e., no flow) boundary conditions, whereas the structurally controlled and complex systems correspond to a domain with both zero and non-zero Neumann boundary conditions. For the tests I ran, only a closed system was considered. The "Pay Thickness Map" input defines the thickness of the reservoir at

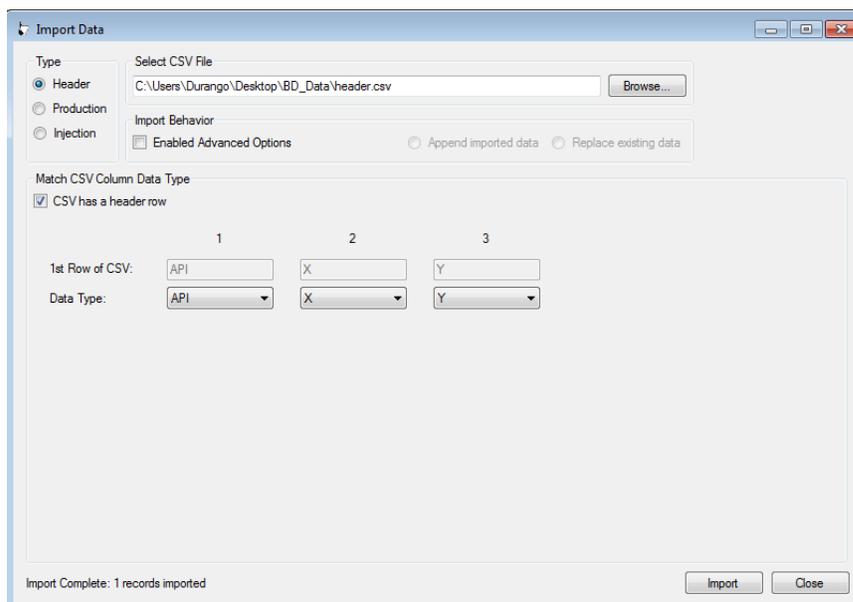


Figure C.2 : Example of importing header data, which defines the locations and APIs of all production and injection wells in the reservoir.

	A	B	C
1	API	X	Y
2	well_0	1275	1275

Figure C.3 : The header data used, for all single producer tests.

every node, in units of feet. Further, the “Flow Path Map” represents the permeability of the reservoir at each node, relative to every other node. This means that ReservoirGrail views a flow path map with a value of 1 at every node equivalently as a flow path map with a value of 10 at every node. The parameters under the “Grid Description” section at the bottom are used to define the spatial discretization of the domain. The “# Columns” and “# Rows” define how many nodes to model in the

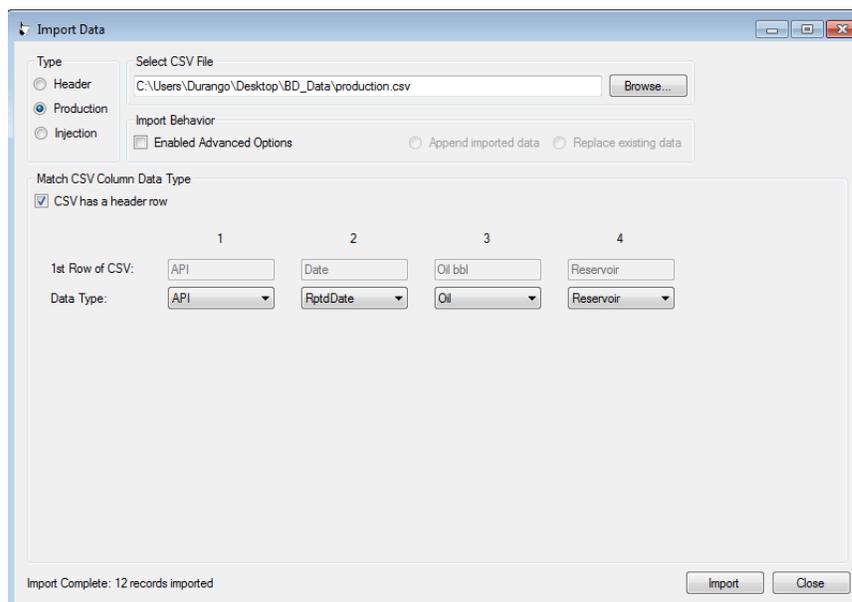


Figure C.4 : Example of importing production well data.

X and Y directions, respectively, the “Minimum X ” and “Minimum Y ” values give the coordinate for the bottom left most node in the domain, and the “Grid Interval” input determines the distance between each node in both the X and Y direction. In this case, a 51×51 grid with the first node at $(25, 25)$ and a spacing of 50 feet between the nodes corresponds to a domain of $(0, 2550) \times (0, 2550)$.

In addition to the above required inputs to this screen, a fault line may also be specified here by giving ReservoirGrail a file path to the fault file, under the “Centerline Fault File” input. Section 3.2.3 discusses how ReservoirGrail interprets fault lines, and figure C.8 shows an example of the fault that I used for my single barrier tests in Section 3.2.3. In this file, the first and second columns represents the X and Y locations, respectively, of the node with a fault passing through it, and the third column represents the fault ID, which is used if multiple different faults are present in the same reservoir.

	A	B	C	D
1	API	Date	Oil bbl	Reservoir
2	well_0	1/1/2014	14476.61	LOWER
3	well_0	2/1/2014	14476.61	LOWER
4	well_0	3/1/2014	14476.61	LOWER
5	well_0	4/1/2014	14476.61	LOWER
6	well_0	5/1/2014	14476.61	LOWER
7	well_0	6/1/2014	14476.61	LOWER
8	well_0	7/1/2014	14476.61	LOWER
9	well_0	8/1/2014	14476.61	LOWER
10	well_0	9/1/2014	14476.61	LOWER
11	well_0	10/1/2014	14476.61	LOWER
12	well_0	11/1/2014	14476.61	LOWER
13	well_0	12/1/2014	14476.61	LOWER

Figure C.5 : The production data used, for all single producer tests.

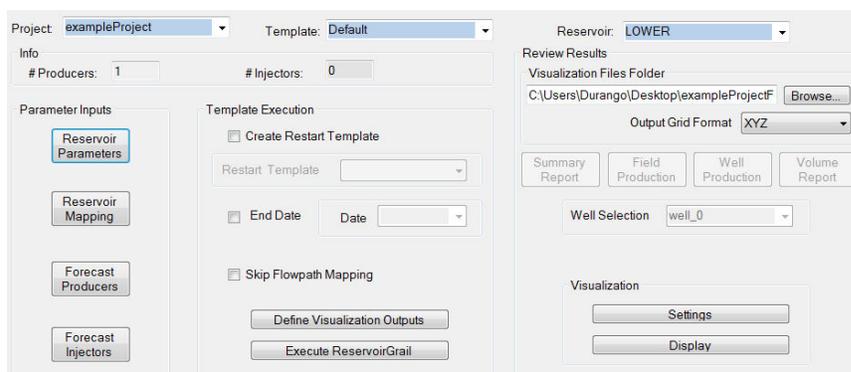


Figure C.6 : ReservoirGrail's home screen.

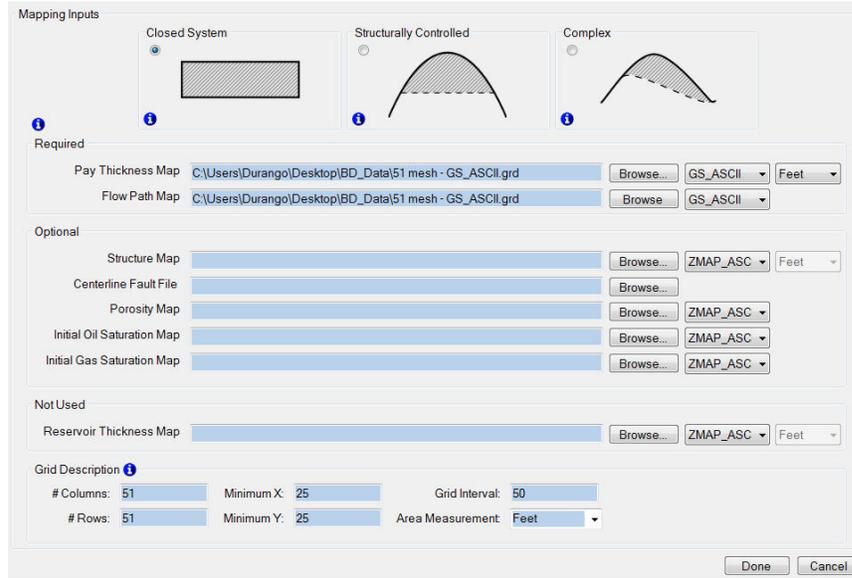


Figure C.7 : Screen where local reservoir features are specified, such as fault lines and permeability changes.

C.3 Defining Reservoir Parameters and Output Files

Going back to the home screen, clicking on the “Reservoir Parameters” tab on the left brings up the screen shown in figure C.9, where the global aspects of the reservoir are specified. The values shown on this screen represent an example of what was used in the tests I ran in this thesis; the only parameter that was changed between simulations was the Equalization Rate, as discussed in detail in Sections 1.2.3 and 2.3. The two parameters in the upper right, under the “Initial Pressure” section, are used to, unsurprisingly, calculate the reservoir’s initial pressure, by multiplying the “Reservoir TVD from Surface” by the “Pressure Gradient”. For the example shown in figure C.9, this value is given as

$$p_{\text{init}} = \text{Reservoir TVD from Surface} \times \text{Pressure Gradient} = 5000\text{ft} \times 0.45 \frac{\text{psi}}{\text{ft}} = 2250\text{psi.}$$

```
575 425 1
575 475 1
575 525 1
575 575 1
575 625 1
575 675 1
575 725 1
575 775 1
575 825 1
575 875 1
575 925 1
575 975 1
575 1025 1
575 1075 1
575 1125 1
575 1175 1
575 1225 1
575 1275 1
575 1325 1
575 1375 1
575 1425 1
575 1475 1
575 1525 1
575 1575 1
575 1625 1
575 1675 1
575 1725 1
575 1775 1
575 1825 1
575 1875 1
575 1925 1
575 1975 1
575 2025 1
575 2075 1
```

Figure C.8 : The fault line data used, for the single barrier tests.

Finally, going back to the home screen in figure C.6, clicking on the “Define Visualization Outputs” tab towards the bottom brings up the screen shown in figure C.10. In order to have ReservoirGrail produce the TransHPV files discussed in Section 2.4, the “Hydrocarbon Pore Volume” box must be checked.

Template: Default Reservoir: LOWER

Fluid Model

Primary Reservoir Fluid: Oil Gas

Gas Cap

Fluid Contacts

Gas/Oil Contact: 0 ft

Oil/Water Contact: 0 ft

Fluid Properties

Bo: 1 rb/stb

1/Bg: 1 scf/rcf

Porosity

Porosity: 1 (decimal)

Initial Oil Saturation

So: 1 (decimal)

Initial Gas Saturation

Sg: 0 (decimal)

Initial Pressure

Reservoir TVD from Surface: 5000 ft

Pressure Gradient: 0.45 psi/ft

Advanced Settings

Waterflood Relative Perm Factor: 1

Equalization Rate: 0.16

Residual Saturations

Input Fraction of Pore Space OR Input Cell Recovery Factor

Primary Residual Oil (Sor1): 0 (decimal)

Secondary Residual Oil (Sor2): 0 (decimal)

Residual Gas (Sgr): 0 (decimal)

Irreducible Water (Swir): 0 (decimal)

Oil Primary Cell Recovery Factor (CRF1): 1 (decimal)

Oil Secondary Cell Recovery Factor (CRF2): 1 (decimal)

Figure C.9 : Screen where global reservoir parameters are specified, such as porosity and initial oil and gas saturation.

Oil Dynamics

Total Oil In Place

Primary Recoverable Oil Volume

Secondary Recoverable Oil Volume

Oil Saturation

Gas Dynamics

Total Gas In Place

Recoverable Gas In Place

Other Dynamics

Pseudo Pressure

Hydrocarbon Pore Volume

Water Saturation

Flooded Pore Volume Ratio

Figure C.10 : Options to determine what outputs ReservoirGrail generates for the simulation.

C.4 ReservoirGrail Flowchart

This section is intended to serve as a high-level abstraction of ReservoirGrail's algorithm, described in Section 1.2.3. At the beginning of every time-step, ReservoirGrail starts in the well phase shown in figure C.11, and after meeting the appropriate stopping criterion, moves to the production phase in figure C.12. Once stopping criterion are met in the production phase, the time-step is incremented and ReservoirGrail starts over at the well phase, until all time-steps have been simulated.

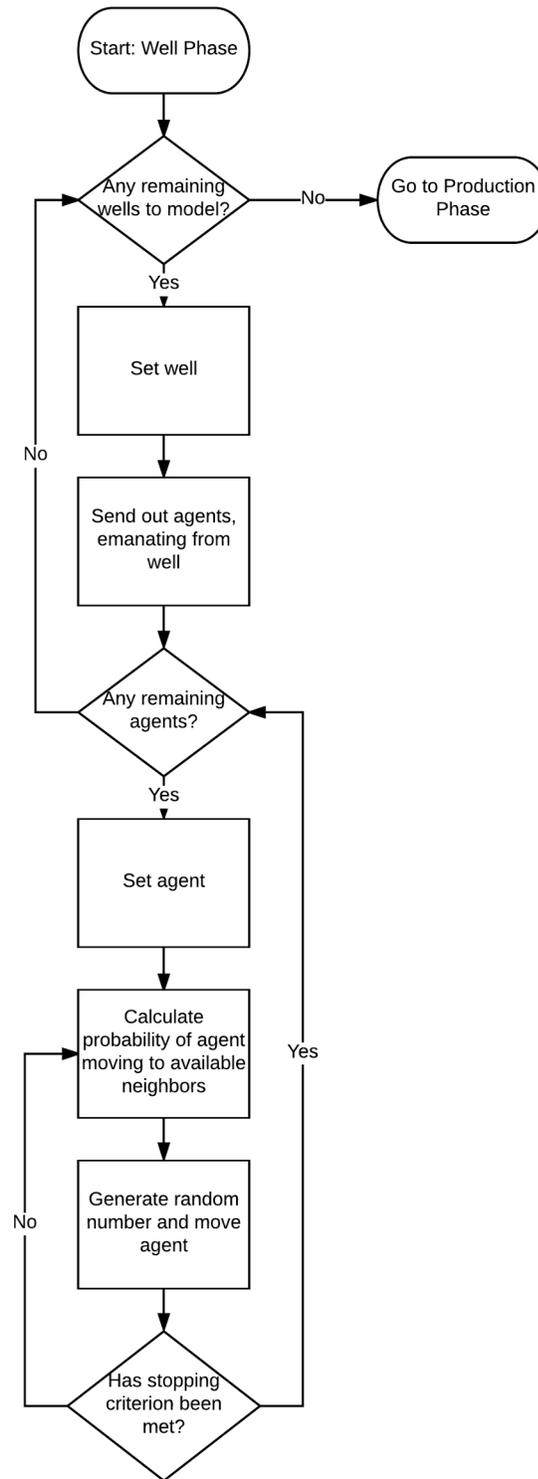


Figure C.11 : A flowchart of the well phase of ReservoirGrail's algorithm, which is used to send out agents and record their visits across the domain.

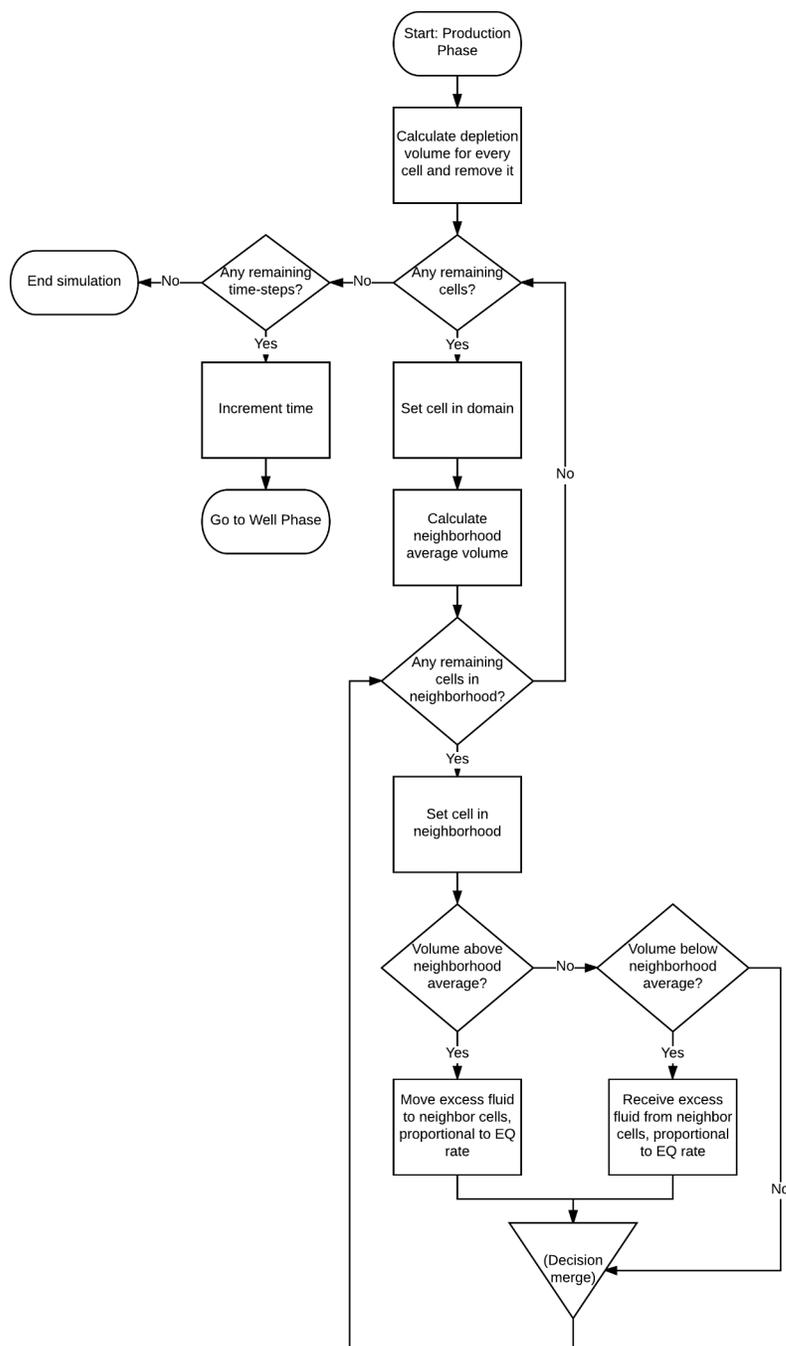


Figure C.12 : A flowchart of the production phase of ReservoirGrail's algorithm, which is used to remove the appropriate amount of hydrocarbons from each cell and equalize the remaining oil, gas, and water as necessary.

Bibliography

- [1] Trangenstein, John A., & Bell, John B. (1989). Mathematical Structure of Compositional Reservoir Simulation. *SIAM Journal on Applied Mathematics*, 10(5), 817-845. Retrieved from <http://epubs.siam.org/doi/abs/10.1137/0910049>
- [2] Zhang, Q., & Shu, C. (2010). Stability Analysis and A Priori Error Estimates of the Third Order Explicit Runge-Kutta Discontinuous Galerkin Method for Scalar Conservation Laws. *SIAM Journal on Applied Mathematics*, 48(3), 1038-1063. Retrieved from <http://epubs.siam.org/doi/abs/10.1137/090771363>
- [3] North, M. (2014). A Theoretical Formalism for Analyzing Agent-Based Models. Springer Open. Retrieved from <https://casmodeling.springeropen.com/articles/10.1186/2194-3206-2-3>
- [4] Conte, R., & Paolucci, M. (2014). On Agent-Based Modeling and Computational Social Science. *Frontiers Journal*. Retrieved from <http://journal.frontiersin.org/article/10.3389/fpsyg.2014.00668/>
- [5] Azimi, M., Jamali, Y., & Mofrad, M. (2011). Accounting for Diffusion in Agent Based Models of Reaction-Diffusion Systems with Application to Cytoskeletal Diffusion. *PLOS ONE Journal*. Retrieved from <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0025306>
- [6] Riviere, B., & Wheeler, M. F. (2000). A Discontinuous Galerkin Method Applied to Nonlinear Parabolic Equations. *Discontinuous Galerkin Meth-*

- ods, 231-244. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-59721-3_17
- [7] Epstein, J., & Axtell, R. (October 11, 1996). *Growing Artificial Societies: Social Science From the Bottom Up*, 224. Brookings Institution Press. ISBN 978-0-262-55025-3.
- [8] Jenny, P., Lee, S.H., & Tchelepi, H.A. (2004). Adaptive Multiscale Finite-Volume Method for Multiphase Flow and Transport in Porous Media. *SIAM Journal on Applied Mathematics*, 3(1), 50-64. Retrieved from <http://epubs.siam.org/doi/abs/10.1137/030600795>
- [9] Kurve, A., Kotobi, K. & Kesidis, G. *Complex Adaptive System Modeling* (2013), 1:12. DOI:10.1186/2194-3206-1-12. Retrieved from <https://link.springer.com/article/10.1186/2194-3206-1-12>
- [10] Fallah, N.A., Bailey, C., Cross, M., & Taylor, G.A (1999). Comparison of finite element and finite volume methods application in geometrically nonlinear stress analysis. *Applied Mathematical Modeling*, 24(2000), 439-455. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0307904X99000475>
- [11] Mi, Q., Riviere, B., Clermont, G., Steed, D. L., & Vodovotz, Y. (2007). Agent-based model of inflammation and wound healing: insights into diabetic foot ulcer pathology and the role of transforming growth factor- β 1. *Wound Repair and Regeneration*, 671-682. doi:10.1111/j.1524-475X.2007.00271.x
- [12] Folick, V. A., An, G. C., & Orosz, C. G. (2007). *The Basic Immune Simulator: An agent-based model to study the interactions between innate and adaptive*

- immunity. *Theoretical Biology and Medical Modeling*. doi:10.1186/1742-4682-4-39
- [13] Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 151-162. doi:10.1057/jos.2010.3
- [14] Hazhir Rahmandad, John Sterman, (2008) Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. *Management Science* 54(5):998-1014. <https://doi.org/10.1287/mnsc.1070.0787>
- [15] Seekhao, N., Shung, C., Jaja, J., Mongeau, L., & Li-Jessen, N. Y. (2016). Real-Time Agent-Based Modeling Simulation with in-Situ Visualization of Complex Biological Systems: A Case Study on Vocal Fold Inflammation and Healing. 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). doi:10.1109/ipdpsw.2016.20
- [16] Fekete Associates Inc. (2012). Oil Compressibility. Retrieved September 21, 2017, from http://www.fekete.com/SAN/TheoryAndEquations/WellTestTheoryEquations/Oil_Compressibility.htm
- [17] (2012, May 20). Retrieved October 04, 2017, from <https://www.youtube.com/watch?v=PN4GflvJr7c>
- [18] GrailQuest. (2011, November 29). Patent US8069018 - Computer system and method for modeling fluid depletion. Retrieved October 04, 2017, from <https://www.google.com/patents/US8069018>
- [19] ReservoirGrail. (2017). Retrieved October 04, 2017, from <http://reservoirgrail.com/>

- [20] Personal correspondence with Ben Bahorich, former employee at GrailQuest.
- [21] Li, J., & Rivière, B. (2016). Numerical Modeling of Miscible Viscous Fingering Instabilities by High-Order Methods. *Transport in Porous Media*, 113(3), 607-628. doi:10.1007/s11242-016-0715-x