NTNU – Trondheim
Norwegian University of
Science and Technology

Microbial Enhanced Oil Recovery

Modeling and Numerical Simulations

Aleksander Amundsen

Applied and Engineering Mathematics	
Submission date:	July 2015
Supervisor:	Helge Holden, MATH
Co-supervisor:	Sidsel Marie Nielsen, Danmarks Tekniske Universitet Knut-Andreas Lie, SINTEF Xavier Raynaud, SINTEF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Microbial Enhanced Oil Recovery: Modeling and Numerical Simulations

Aleksander Amundsen

July 16, 2015

Abstract

This thesis examines the process by which microbes are used to enhance oil recovery from subsurface reservoirs. A brief introduction to reservoirs is given and the possible effects of microbes are explained. A model is developed combining porous media flow and microbial kinetics. The model is then used to run simulations in conjunction with the MATLAB Reservoir Simulation Toolbox from SINTEF (Stiftelsen for Industriell og Teknisk Forskning). Microbial enhanced oil recovery (MEOR) is simulated by injecting a microbe and nutrient mix into the reservoir. The microbes consume the nutrients and then reproduce and create metabolites according to the Monod model of bacteria. The metabolites are simulated to be either surfactant or polymer which are both beneficial to oil recovery. Surfactant reduces the residual oil level while polymer increases the volumetric sweep of the reservoir. Langmuir equilibrium adsorption is used to model the formation of biofilm. This results in more metabolites being produced nearer the injection sites. An attempt to recreate the results of Nielsen et al. (2010) [16] and Lacerda et al. (2012) [7] is made. Differences in the results are highlighted and explained. Further testing is done to examine the impact of a high permeability thief zone to the model. The thesis concludes that the model and its implementation have been successful so far though more testing should be done.

Preface

This work fulfills the Master's thesis requirements of the N5TeAM Master's Programme in Applied and Engineering Mathematics, Computational Mechanics study track. The partner universities attended were the Technical University of Denmark (DTU), and the Norwegian University of Science and Technology (NTNU). The thesis is being written at NTNU with Professor Helge Holden as the official advisor. He put me in contact with Knut-Andreas Lie, Chief Scientist, SINTEF, Department of Applied Math, who suggested the thesis topic. He, along with his colleague Xavier Raynaud, helped me learn about oil recovery modeling and how to use the MATLAB Reservoir Simulation Toolbox developed by SINTEF. They also put me in contact with Sidsel Marie Nielsen, post-doctoral researcher at the DTU Department of Chemical and Biochemical Engineering, who is my DTU advisor. Together, they have provided helpful insight and advice on how to complete the project.

I would like to specifically thank my brother, Kris Amundsen, for taking time out of his vacation to read my thesis and provide me with feedback. Even though it is currently 49 hours before my deadline and I have not received any of his notes yet. Irregardless, he had to read it which deserves some credit. I would also like to thank Katiana Efstratiou. She helped me whenever I had basic questions about chemical engineering that I was somehow unable to find answers to on the internet. And I would like to thank Hager Debech for generally being supportive and listening to me complain when I needed to.

Contents

Chapter 1

Introduction

While renewable energy sources are becoming increasingly attractive both from an environmental and innovation viewpoint, the need for oil will remain for the foreseeable future. Not only as an energy source, but for useful petrochemicals such as plastics as well. Intuitively, as fewer new discoveries of large reservoirs are made, it is desirable to accomplish recovery in increasingly efficient ways. This includes maximizing production and minimizing costs. Subsurface reservoirs can contain various types of matter such as gases, oils, and water. While microbes may also be used to protect water in aquifers [22], the focus of this work is how they may be used to improve oil recovery. Recovering oil from subsurface reservoirs is done in multiple stages. Initially, there exists intense pressure in the reservoir, so when a production well is built, the oil flows to the surface through natural driving forces. This is called primary recovery. Afterwards, injection wells begin pumping gas or water into the reservoir in order to maintain the pressure and keep the oil flowing. This method, referred to as secondary recovery, does not succeed in flushing all the oil from the reservoir due to several reasons which will be presented later [8]. This is when enhanced oil recovery (EOR) techniques can be implemented to keep the well operational. There are several different technologies that can be effective in increasing production. Polymer injection mixes a polymer compound with water to increase the viscosity which sweeps the reservoir better, forcing more oil to flow. Thermal processes in which steam is injected or controlled combustion occurs in the reservoir, are used in order to reduce oil viscosity. Surfactant may be injected which can improve the displacement of small quantities of oil trapped in the pores. Alternating injections of water and gas can reduce the mobility of the gas and increase the sweep in the reservoir. This study's main concern is microbial enhanced oil recovery (MEOR). This is the process in which microbes and nutrients are injected into the reservoir to improve recovery. The microbes consume the nutrients to multiply and produce beneficial metabolites. They can produce biopolymers, biosurfactants, gases, and acids which can all contribute to improved recovery. They may also form biofilm that clogs pores which can also benefit overall recovery. The following list of advantages to MEOR was presented in a paper by Lazar et al. (2007) [10],

1. The injected bacteria and nutrient are inexpensive and easy to obtain and handle in the field.

2. Economically attractive for marginally producing oil fields; a suitable alternative before the abandonment of marginal wells.

3. According to a statistical evaluation (1995 in U.S.), 81% of all MEOR projects demonstrated a positive increase in oil production and no decrease in oil production as a result of MEOR processes.

4. The implementation of the process needs only minor modifications of the existing field facilities. It is less expensive to install and more easily applied than another EOR method.

5. The costs of the injected fluids are not dependent on oil prices.

6. MEOR processes are particularly suited for carbonate oil reservoirs where some EOR technologies cannot be applied with good efficiency.

7. The effects of bacterial activity within the reservoir are magnified by their growth whole, while in EOR technologies the effects of the additives tend to decrease with time and distance.

8. MEOR products are all biodegradable and will not be accumulated in the environment, so environmentally friendly.

Despite the advantages, the oil industry typically prefers other EOR technologies. Statoil has stated that they believe they are the only company currently using MEOR in an offshore field [19]. The complexity of the MEOR processes is also a disadvantage. Since there are several reactions occurring that are specific only to MEOR, commercial reservoir simulators, such as Schlumberger's ECLIPSE, do not include MEOR functionality. The overarching goal of this work is then to develop and implement an MEOR simulation model which is customizable and functional. The process is to be simulated using the MATLAB Reservoir Simulation Toolbox (MRST) developed by SINTEF [12]. This is open source code intended to simulate both single and multi-phase flow in porous media with the help of automatic differentiation. New MATLAB scripts are developed to simulate the effects of MEOR. The model and its implementation in MATLAB will be tasked with recreating the results from two previous works simulating MEOR. The first is a biosurfactant study done by Nielsen et al. (2010) [16]. The other, a biopolymer study done by Lacerda et al. (2012) [7]. The model is then tested to examine how an area of high permeability would effect fluid flow and recovery.

Chapter 2

Reservoir Introduction

Before modeling considerations may be formulated, a general introduction to reservoirs and oil recovery will be useful. The reservoirs are formed geologically over the course of millions of years [14]. The reservoirs are not simply a void in the rock where the liquid accumulates. They are comprised of several layers of permeable porous rock whose properties can be quite varied. Different layers of rock are formed by different geological processes and can be read about further in a introductory geology textbook [6]. The reservoirs contain hydrocarbons which are the product of decomposed organic life [14]. The reservoirs can cover hundreds of square kilometers, though they have a relatively thin depth [15]. Although there may be billions of cubic meters of oil in a reservoir, only a fraction of that oil is attainable through standard water flooding recovery procedures. Recovery of only 35% of the original oil in place is considered an average result [14]. Part of the reason for this is that reservoirs are not homogeneous. The pore network within the rock is quite extensive and not optimized for fluid flow. The porosity of a reservoir ϕ, is a measure of its void space given as a dimensionless ratio

$$\text{porosity} = \phi = \frac{\text{void volume of rock}}{\text{total volume of rock}}.$$

Within traditional oil reservoirs, this number typically resides between 0.1 and 0.4 [12]. The porosity changes based on the type of rock formation within the reservoir. The permeability of a rock is a measure of the ease with which liquid may pass through these pores. It is measured in a unit called darcys [9]. As different liquids may perceive a medium to be more or less permeable, the darcy is a reference to water on the Earth's surface. One darcy is approximately 9.87×10^{-13} m^2 and is also the permeability value of sand [9]. This is because Henry Darcy, the unit's namesake, preformed experiments on the speed with which water would pass through a column of sand in the 1850s [1]. A large porosity value does not automatically guarantee a high permeability level. For instance, a rock could have many very small pores unsuitable for fluid flow, or, the fluid could be flowing horizontally while all the pores are vertical. These examples are not entirely plausible but do illustrate the complexity of the relationship between porosity and permeability. In fact, the porosity value in a reservoir may barely vary while the permeability spikes significantly [12]. There are also cases where oil is contained in nearly impermeable rock which previously meant it was irrecoverable. However, with hydraulic fracturing, this is no longer the case [14]. Because the permeability can differ greatly within a reservoir, so called thief zones are created. This is a phenomena in which the injected fluid develops channels of flow through highly permeable areas, the thief zones, hence avoiding less permeable areas. The oil in these areas remains unmolested and will not flow towards a production well. This problem may be addressed by reducing the permeability of the thief zone by either changing the porosity of the rock with some clogging effect, or the properties of the fluid with a chemical mixture. However, even if the reservoir has a homogeneous permeability, a substantial

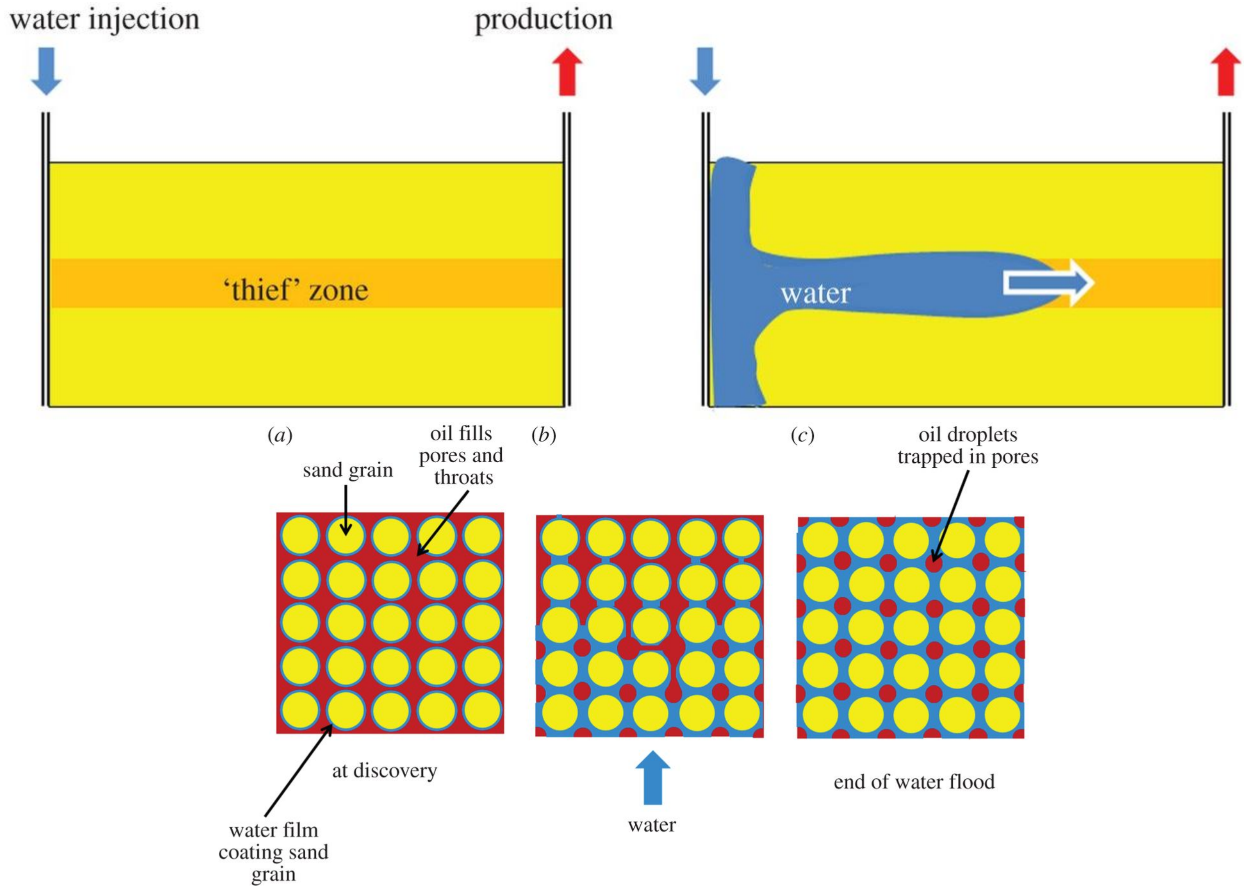


Figure 2.1: Visualization of thief zones (above) and capillary trapping (below) from Muggeridge et al. (2013) [15]. In both cases, a substantial portion of the oil is left behind and does not flow towards the production well.

portion of the oil will still remain after flooding due to capillary trapping. This is when the interfacial tension between the two phases is such that the oil is rendered immobile. Because the water and oil do not mix to form an emulsion, there is a force acting between the surfaces of the fluid where they meet called interfacial tension (IFT). There is a different pressure in each of the phases and the difference is called the capillary pressure [1]. The higher the IFT, the higher the capillary pressure. High capillary pressure leads to capillary trapping, where the oil is being pressed upon to move from multiple sides by the water and is in a state of equilibrium. This includes lone droplets of oil, or pores full of oil with water blocking both their entrance and exit. Simple graphical explanations of both thief zones and capillary trapping can be seen in Figure 2.1 taken from Muggeridge et al.(2013) [15]. We see the injected water traveling through the thief zone as it is the path of least resistance, and avoiding the other areas. In the lower figure, we see how some oil can be separated and left behind during water flooding because of capillary trapping. Tertiary recovery, or EOR, is the name given to injection processes that try to mitigate the effects of thief zones and capillary trapping. While the naming conventions of primary, secondary, and tertiary recovery indicate a sequential process, this is in fact a misnomer. There need not be any separation of the recovery stages though this is often the case. A production well is created first so that revenue can be made on the operation, with injection wells built after. Typically, due to the increased cost of EOR, water or gas flooding is continued until production levels begin to decline. This is when most EOR practices are implemented, even though they would have also been effective earlier in the operation [15].

2.1 Enhanced Oil Recovery

There are several different methods of EOR which can mostly be divided into the categories of thermal, chemical, and gas [14]. Thermal EOR is achieved in a variety of ways. This includes the injection of steam or hot water, and the process of in situ combustion where controlled ignitions occur inside the reservoir. The idea behind thermal EOR processes is that the added heat will reduce oil viscosity which improves the flow. This is because the flow rate of a fluid is dependent on its viscosity which appears later in equation (3.3). Steam injection is quite effective with typical increased recovery between 50 and 65 percent of the original oil in place [8]. It is however a difficult method to employ as it requires boiling large quantities of water in the field. The water used offshore is typically the surrounding seawater which generates a less effective steam and the fuel used to generate the steam is the produced crude oil which leads to air pollution [8].

Chemical EOR includes the use of polymers, surfactants, and alkalines to assist in extraction. The chemicals alter the physical properties of the water, improving the displacement of oil. Polymer increases the viscosity of water which increases the volumetric sweep of the water flood. The result is that thief zones are no longer problematic and a larger portion of oil is displaced [14]. The surfactants injected strive to lower the IFT between the water and oil phases. The surfactant is a monomer with hydrophilic and lypophilic ends [8]. This duality causes the surfactant to accumulate at the interface of the two phases which reduces the IFT. Alkaline flooding seeks to create surfactants in situ to reduce IFT. A common practice is to combine these methods into alkaline surfactant polymer (ASP) flooding [15]. This method has the benefits of both increasing sweep and reducing IFT. Some of the main difficulties with employing chemical EOR are the logistics. For offshore platforms, the chemicals must first be delivered and then stored on a structure where limited space is already an issue.

While injecting a gas such as CO_2 in a reservoir is a secondary recovery technique, a combination of oil and gas is considered an enhanced technique. This can be performed simultaneously or in an alternating fashion. The advantages are that vertical sweep is improved as the dense water tends towards the bottom of the reservoir while gas tends towards the top. It also reduces the thief zone effect when compared with solely gas injection [15]. This is because the presence of a mobile water phase effectively reduces the permeability for the gas. Consequently, alternating between injection types needs to follow a specific schedule to remain effective. This is a challenge as platform conditions may dictate that the physical switching of the injection type of a well is not a priority [15].

These three processes constitute the main effects and advantages of EOR. This is why there is such interest around MEOR, becuase it has the ability to achieve the same results when used effectively and properly.

2.1.1 Microbial Enhanced Oil Recovery

Enhancing oil recovery with the use of microbes is a concept that was first presented in the 1920s [10]. The essential idea is that the microbes consume nutrients and reproduce while beneficial metabolites such as surfactants and polymers are a biological byproduct of this process. There are three methods of MEOR utilization. The first injects a solution of microbes and nutrients into the well. The second injects only nutrients in hopes of activating the in situ microbes. The third involves developing the microbes outside of the reservoir and only injecting their metabolites into the well. The latter method is not considered in this work and will not be discussed or mentioned further as it falls outside of the primary focus of this study. The aggregation of several microbes in the reservoir results in biofilm that can clog the pore network. While at first glance one may consider this to be a negative aspect, selective plugging can result in a lower permeability of thief zones and thus increase the sweep of the water flood. As the microbes are organic, a great deal of research must be done to ensure the reservoir is capable of sustaining the population. This includes factors such as temperature, pH, salinity, and pore size. If the temperature is too high, the microbes will not function properly meaning no reproduction, production of metabolites, or formation of biofilm. A low pH value hinders the ability of microbes to reproduce as does the salinity of the water. The size of bacteria allows for clogging of pores but ideally this occurs somewhere other than the near-well area. Pore diameter should ideally be in the range of 6-10 μm with the bacteria diameter closer to 2 μm [17]. An effort should also be made to determine if there are any microorganisms that the injected microbes will have to compete with for nutrients.

There is another factor that hinders the efficacy of MEOR and chemical EOR, and that is adsorption. This is a process in which a substance adheres to the pore walls on an atomic level, through typically chemical or electrical attraction. This occurs mostly with consideration to the microbes but also with polymers and surfactants to a lesser extent [17]. For MEOR it produces a similar effect to biofilm formation though is less desirable as it behaves without discretion. The concern with polymer and surfactant, along with their biologically produced counterparts, is that they are no longer flowing and improving oil recovery when adsorbed and effectively become worthless. It is evident that there are several processes occurring simultaneously which are influenced by many factors. It is a daunting task to encapsulate them all into a single model and therefore there will be simplifications made. We will first begin by detailing traditional reservoir modeling.

Chapter 3

Reservoir Modeling

The first consideration made when modeling flow in a reservoir is the incredible difference in scales. There is the microscopic scale where the fluid flows through the pores, and there is the macroscopic scale in which the reservoir structure may cover several square kilometers. Simplifications must be made to reconcile these differences. The most important being a conceptual shift to continuum mechanics to describe the actions occurring on the microscopic scale. Rather than being concerned with how fluid winds around in the pore network, the use of a representative elementary volume (REV) is employed. The microscopic traits and action in the reservoir are now reformed into averages of a larger volume. This change allows for better description of the reservoir and the use of conservation principles to describe the dynamics of the recovery. Conservation will serve to describe not only fluid flow, but also the transport of microbes, nutrients, and metabolites. The change of mass inside the elementary volume V will be calculated as

$$\left\{\begin{matrix}\text{Change in}\\ \text{mass of}\\ V\end{matrix}\right\} = \left\{\begin{matrix}\text{Mass}\\ \text{transported}\\ \text{into } V\end{matrix}\right\} - \left\{\begin{matrix}\text{Mass}\\ \text{transported}\\ \text{out of } V\end{matrix}\right\} + \left\{\begin{matrix}\text{Source}\\ \text{inside}\\ \text{of } V\end{matrix}\right\} \tag{3.1}$$

The source term will account for the consumption of nutrients and the production of bacteria and their metabolites. Modifications for compressiblity will be made later.

3.1 Rock Modeling

The porosity and permeability serve to define the rock properties in the reservoir. As the individual pore structure is too complex, an average porosity value is of interest. The concept of porosity, ϕ, and permeability, $\mathbf{K}$, was introduced earlier as was their complicated connection. The Kozeny-Carman equation is often used to interpret their relationship,

$$K = \frac{1}{8\tau A_v^2}\frac{\phi^3}{(1-\phi)^2}.$$

In this equation, A_v is the ratio of the internal rock surface area to volume, while τ is the tortuosity which is a ratio of the length of a pore to the distance between its endpoints,

$$A_v = \frac{\text{Total surface area of rock}}{\text{Total volume of rock}}$$
$$\tau = \frac{\text{Pore length}}{\text{Distance between pore entrance and exit}}$$

These measurements are not feasible to obtain and map for an entire reservoir. However, the Kozeny-Carman equation does serve to acquire estimates of permeability, or porosity if the former is known [8]. Another limitation of the equation is that the resulting value is a scalar, K, while permeability in a reservoir model is more accurately represented by a tensor, **K**. The reason being that fluid does not flow identically in all directions. The permeability is most often obtained through measurements of macroscopic flow [12]. It is important to mention that there are multiple types of permeability connected to reservoir modeling. **K** is referred to as the absolute permeability. When we expand into a simulation with more than one fluid, the concept of relative permeability arises from the need to alter the absolute permeability to describe each fluid according to its properties.

3.2 Porous Media Flow

We have taken advantage of the REV to simplify the properties of the rock, now it is time to consider the flow of fluid. The main fluid properties of importance are the density, ρ, viscosity, η, and saturation, S. It is also common to represent viscosity with μ, however that is being reserved for later to describe bacterial growth rates. The saturation will be used to distinguish between the composition of fluid in the REV and presented as a ratio

$$S_i = \frac{\text{Volume of fluid } i \text{ in } V}{\text{Total volume of fluid in } V}.$$

The subscripts o and w will serve to represent oil and water respectively. Additionally, as this work will be confined to strictly oil and water phases, the relationship $S_o = 1 - S_w$ may be used. The mass of a fluid in V is now a product of the saturation and density of the fluid, as well as the porosity and volume of V. To calculate the amount of fluid coming in and out of the volume, the flux, the velocity with which it is traveling must be determined.

3.2.1 Darcy's Law

Fluid flows in a reservoir because of an ambient pressure gradient created by the wells, fluid buoyancy, and surrounding aquifers. This flow is influenced by the fluid properties, permeability of the rock, and gravity. Again, as the pore structure is too small and complex, a macroscopic velocity is of interest. This is a vector representing the effective direction and speed of the fluid in the pores contained in the REV. And so returns Henry Darcy with his model for calculating macroscopic velocity in porous media, $\vec{u}$, aptly named Darcy's Law

$$\vec{u} = -\frac{1}{\eta}\mathbf{K}(\nabla p - \rho g \vec{e}_z). \tag{3.2}$$

Here p represents pressure, g the gravitational constant, and $\vec{e}_z$ is the basis vector for the vertical axis. This equation was developed by Darcy who observed water flowing through sand. Though the relationship was found experimentally, it can also be derived theoretically from the Navier-Stokes equations [24]. What equation (3.2) is describing, is that the macroscopic velocity, $\vec{u}$, is inversely proportional to the viscosity of the phase, η, in the direction opposite increasing pressure, with modifications made for both gravity and absolute permeability of the rock, **K**. All necessary elements are now present to model fluid conservation in a reservoir.

3.3 Conservation of Mass

To calculate mass conservation with (3.1), we use the REV as our computational domain, Ω, and evaluate volume integrals and line integrals. The equation for mass conservation of a single phase is then,

$$\frac{\partial}{\partial t}\int_{\Omega}\phi\rho\,dV + \int_{\partial\Omega}\rho\vec{u}\cdot\vec{n}\,ds = \int_{\Omega} q\,dV,$$

where $\vec{n}$ is the outward normal of the surface of the region and q is a source term. In reservoir modeling, the source term represents the injection and production wells and therefore is only nonzero in an REV containing a simulated well. Hence, $q < 0$ indicates a production well that is removing mass from the system and $q > 0$ represents injection wells adding mass. Using the divergence theorem, or Gauss' theorem, we observe that the flux through the surface is equal to the integral of the divergence inside the volume.

$$\int_{\Omega}\left(\frac{\partial}{\partial t}\phi\rho + \nabla\cdot(\rho\vec{u})\right)dV = \int_{\Omega} q\,dV.$$

As the region is arbitrary, mathematically we may drop the integrals by shrinking the REV to an infinitesimal size. We are then left with a partial differential equation to model conservation,

$$\int_{\Omega}\left(\frac{\partial}{\partial t}\phi\rho + \nabla\cdot(\rho\vec{u})\right)dV = \int_{\Omega} q\,dV$$
$$\frac{\partial}{\partial t}\phi\rho + \nabla\cdot(\rho\vec{u}) = q$$

Combining equations (3.2) and (3.1), our system for modeling fluid flow through porous media is

$$\begin{aligned}\vec{u} &= -\frac{1}{\eta}\mathbf{K}(\nabla p - \rho g\vec{e}_z)\\ \frac{\partial}{\partial t}\phi\rho + \nabla\cdot(\rho\vec{u}) &= q.\end{aligned} \tag{3.3}$$

This equation is however only formulated for a single fluid and we are of course interested in modeling more than just one phase.

3.4 Multi-Phase Flow

While previous sections have referred to fluid flow, conservation is also valid for gaseous flow as well. This is significant as gas may be used during injection or already be present in the reservoir. The term phase refers to any flowing medium in the reservoir with distinct properties. In order to fully capture flow in a production reservoir, multi-phase models must be used. Because oil and water have different values for density and viscosity, and furthermore do not mix, these must be modeled separately using different values. This is achieved by the introduction of the saturation values to (3.3) and making the computations for each phase individually,

$$\begin{aligned}\vec{u}_i &= -\frac{1}{\eta_i}\mathbf{K}(\nabla p - \rho_i g\vec{e}_z)\\ \frac{\partial}{\partial t}\phi\rho_i S_i + \nabla\cdot(\rho_i\vec{u}_i) &= q_i.\end{aligned} \tag{3.4}$$

The rock properties and pressure values are not affected. Any substance injected into the reservoir along with the water phase will be similarly transported in the reservoir. These substances will be represented as a concentration of mass to volume, kg/m^3. To realistically model fluids in a reservoir, important physical considerations must be made. For instance, the fact that fluid densities at surface conditions are different than inside the reservoir and also different phases experience different permeability conditions.

3.4.1 Compressibility of Fluids

The concept of compressibility must be applied to the reservoir model. Because fluids in real life are compressible, their density will be different inside a reservoir from what is measured on the surface. This is typically reconciled using what is referred to as a formation volume factor, β_i, where i is the relevant phase [4]. The formation volume factor is a ratio comparing the volume of the fluid at surface conditions to the same volume inside the reservoir. This is a measure of compressibility that is affected by pressure and temperature and, in the case of oil, how volatile the oil is. A volatile oil will experience a large amount of shrinkage due to high levels of gaseous hydrocarbons in the system [1]. For modeling an oil that is not affected by pressure and temperature change, this factor is 1. That number increases for more volatile hydrocarbons. This enables the following relationship to be defined and taken advantage of

$$\rho_{sc} = \rho\beta.$$

This relationship is used for all phases though the subscript was withheld for clarity and the subscript sc indicates typical surface conditions. Equation (3.4) can be rewritten with the formation volume factor to obtain,

$$\frac{\partial}{\partial t}\left(\frac{S_i\phi\rho_{sc,i}}{\beta_i}\right) + \nabla\cdot\left(\frac{\vec{u}_i\rho_{sc,i}}{\beta_i}\right) = \tilde{q}_i\rho_{sc,i}$$

where $\tilde{q}_i$ is the modified source term per volume for surface conditions. The surface densities can then be removed from the equation to obtain the specific partial differential equations we are interested in

$$\frac{\partial}{\partial t}\left(\frac{S_w\phi}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w}{\beta_w}\right) = \tilde{q}_w$$
$$\frac{\partial}{\partial t}\left(\frac{S_o\phi}{\beta_o}\right) + \nabla\cdot\left(\frac{\vec{u}_o}{\beta_o}\right) = \tilde{q}_o$$

for water and oil respectively. To introduce a substance that is being transported along with the water, we define c as a concentration of the substance in kg/m^3. Then, c_{inj} is the concentration of the substance being injected. This results in,

$$\frac{\partial}{\partial t}\left(\frac{S_w\phi\, c}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w c}{\beta_w}\right) = \tilde{q}_w c_{inj}$$

to model the transport of the injected substance. The introduction of the formation volume factor has enhanced the fluid model into something more realistic. However, there are more considerations that must be made.

3.4.2 Relative Permeability

There are devices that make it possible to create an accurate geological model of an existing reservoir. It is however, not possible to run any worthwhile simulations on a highly detailed model [12]. The difference in the scales is too great and hence, the computations too demanding. For that reason, simplifications are made in order to achieve useful results. In a physical reservoir, there may be geological fractures and faults that introduce an abrupt change in magnitude of the permeability, or even a layer of virtually impermeable clay. Therefore, it is beneficial to divide the reservoir into several regions during computation. It is desirable to keep the size of these regions reasonably small so differences may be preserved, though still large enough that simulations are computationally reasonable. The absolute permeability of these regions will be averaged values. However, effective permeability is not a static value, it will change depending on the fluid saturation present in the volume. This is where the concept of relative permeability enters the framework. The goal of relative permeability is to model the effects of IFT that occur on the microscopic scale. The relative permeability, denoted kr_i, enters in as a multiplier to the calculation of the velocity $\vec{u}$,

$$\vec{u}_i = -\frac{kr_i}{\eta_i}\mathbf{K}(\nabla p - \rho_i g \vec{e}_z). \tag{3.5}$$

Its purpose is to alter the absolute permeability tensor for multi-phase flow, giving instead an effective permeability and simulating the effect of capillary trapping. If it is not desirable to simulate capillary trapping, the relative permeability is simply the phase saturation, that is $kr_i = S_i$ [8]. The change in relative permeability is a product of the capillary trapping, when oil is trapped in a pore due to the IFT. Essentially, there is a pressure difference between the two phases at their interface, eventually resulting in some oil being rendered immobile as in Figure 2.1. This is often simulated using Corey relative permeabilities [8]. This is a function of the saturation level, S_i, of a phase that restricts the flow at low saturation levels. The other factors affecting the relative permeability are the residual oil level, S_{or}, and the initial saturation level of water, S_{wi}, this is also referred to as the connate water saturation. These values represent the saturation levels at which the respective phase is immobile. Another important value is the relative permeability of the opposite phase at these levels. This is represented as kr_{owi} for the oil relative permeability at initial water saturation, and kr_{wor} for the water relative permeability at residual oil saturation. The equations are then,

$$\begin{aligned} kr_o(S_o) &= kr_{owi}\left(\frac{S_o - S_{or}}{1 - S_{wi} - S_{or}}\right)^n \\ kr_w(S_w) &= kr_{wor}\left(\frac{S_w - S_{wi}}{1 - S_{wi} - S_{or}}\right)^n \end{aligned} \tag{3.6}$$

where n is referred to as the Corey exponent. This alters the curvature of the relative permeability and does not necessarily have to hold the same value for both curves. The relative permeability profiles used in Lacerda et al. (2012) [7] may be observed in Figure 3.1. Explicitly, the equations are

$$\begin{aligned} kr_o(S_o) &= 0.7\left(\frac{S_o - 0.23}{1 - 0.16 - 0.23}\right)^2 \\ kr_w(S_w) &= 0.3\left(\frac{S_w - 0.16}{1 - 0.16 - 0.23}\right)^2 . \end{aligned}$$

This ensures that the oil saturation does not decrease past the residual level. At $S_o = S_{or}$, $kr_o = 0$. When entered into equation (3.5), this results in $\vec{u}_o = 0$. With the velocity being zero, the divergence will be zero as well, hence change to the mass conservation equation (3.4), can only originate from outside sources. The goal of EOR with surfactant is to decrease the IFT and ultimately the residual oil level. Modifications to the model have now been made for compressibility and relative permeability. The current version of the conservation equations for a two-phase oil and water system is now

$$\begin{aligned} \frac{\partial}{\partial t}\left(\frac{S_w\phi}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w}{\beta_w}\right) - \tilde{q}_w = 0 \\ \frac{\partial}{\partial t}\left(\frac{S_o\phi}{\beta_o}\right) + \nabla\cdot\left(\frac{\vec{u}_o}{\beta_o}\right) - \tilde{q}_o = 0 \end{aligned} \tag{3.7}$$

where,

$$\begin{aligned} \vec{u}_w &= -\frac{kr_w\mathbf{K}}{\eta_w}\left(\nabla p - \rho_w g\cdot\vec{e}_z\right) \\ \vec{u}_o &= -\frac{kr_o\mathbf{K}}{\eta_o}\left(\nabla p - \rho_o g\cdot\vec{e}_z\right). \end{aligned} \tag{3.8}$$

With the model for standard two-phase flow developed, we may begin alterations to model MEOR.

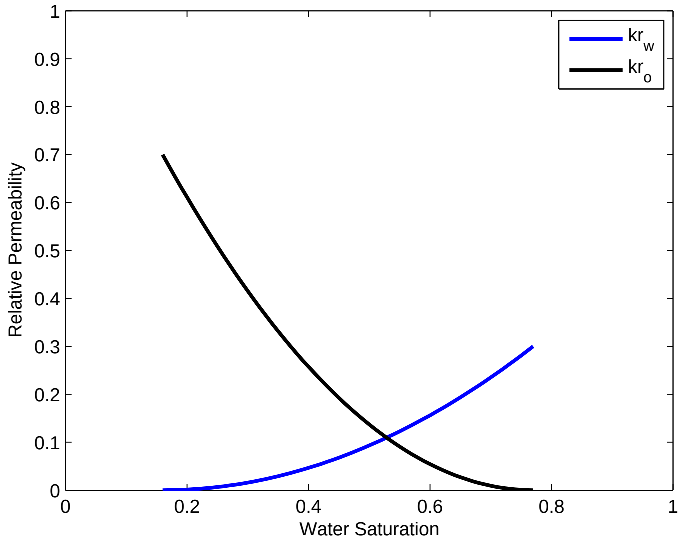


Figure 3.1: The Corey relative permeability curves of [7] with $S_{wi} = 0.16$ and $S_{or} = 0.23$. As the saturation does not exceed beyond the residual values, there are no values for the relative permeability in these areas.

Chapter 4

MEOR Modeling

With equations (3.7) and (3.8) to model two-phase flow in a reservoir, the addition of MEOR mechanisms will now be considered. This will consist of microbes, nutrients and metabolites being transported in the aqueous phase along with the formation of biofilm. The same conservation concepts presented in (3.1) may still be applied. The source terms will now include the reactions due to the microbes. This includes consumption of nutrients, microbial reproduction, and metabolite production. The effects of the metabolite will also be modeled dependent on its type. We choose to only consider biosurfactant and biopolymer at this time as they are the most commonly considered and modeled products [7]. The biosurfactant will reduce IFT and alter the relative permeability curve to reduce the residual oil level. Biopolymer will increase the viscosity of water which improves the displacement of oil in the reservoir. We also examine the potential of the microbes adsorbing to the rock walls and forming biofilm. This biofilm will reduce both the porosity and the permeability of the reservoir which can mitigate the harm from thief zones.

4.1 Microbe Modeling

While the term microbe may be applied to any single-celled organism, what is of specific interest to MEOR is bacteria. Bacteria is highly resilient and can survive extreme conditions. This is evidenced by the existence of bacteria inside reservoirs that haven't had outside exposure since their formation. Bacteria fall into three categories based on whether or not they require oxygen. Aerobic bacteria require oxygen to survive whereas oxygen is in fact toxic to anaerobic microorganisms [17]. Facultative microbes can function as either aerobic or anaerobic. It is possible for aerobic bacteria to survive in a reservoir as long as there is an oxygen supply. The respiration occurs through electron transfer and requires more than H_2O to be effective [13]. While it is not impossible to use aerobic bacteria for MEOR, their use increases the complexity and requires more careful study of the reservoir conditions beforehand. Much research must be done before a specific bacteria species is chosen for MEOR. The choice is influenced by growth rate, the type of nutrient it can use, the desired metabolite produced, and how it forms biofilms. A successful mathematical model of this bacteria must describe all these different processes. Ultimately, the model of microbes must be combined within the framework of the conservation equation (3.1). The microbes will be transported within the reservoir in the water phase, consuming nutrients and producing metabolites as they go.

4.1.1 Monod Model

The Monod equation is an empirical description of bacterial growth based on Michaelis-Menten kinetics [17]. It is used to determine the specific growth rate, μ, of bacteria as a function of the limiting nutrient concentration, N, in kilograms per meter cubed. This is a product of the observed maximum growth rate, μ_{max}, and the half-saturation constant, K. The half-saturation constant is the nutrient concentration level when $\mu = 0.5\,\mu_{max}$. The Monod equation is then,

$$\mu = \mu_{max}\frac{N}{K+N}. \tag{4.1}$$

This growth rate is then multiplied by the concentration of bacteria, B, to be used as the source term in the conservation equation (3.1). In Lacerda et al. (2012) [7], a similar equation is used to describe metabolite production. Like the bacteria, B, and nutrients, N, the metabolites, M, are given in concentrations of kg/m^3. The production rate of metabolites is

$$\mu_m = \mu_{m\,max}\frac{N - N_{\text{crit}}}{K_m + N - N_{\text{crit}}}.$$

This is identical to the form of the Monod equation (4.1) except for the term N_{crit}, which is used to model a need for a minimum concentration of nutrients in order for metabolite production to occur. This rate is then multiplied by the bacteria concentration to determine the production source term of metabolites. However, to now ensure that the nutrients are not being used for both bacterial reproduction and metabolite production at the same time, a yield coefficient, Y_i, is introduced. This is a measurement of how much nutrient is used to arrive at the respective production rates. It is given as a ratio so that $Y_b + Y_m = 1$, and no nutrients are being used twice. Therefore, the reaction terms of bacteria and metabolites are

$$R_b = \mu_b B Y_b$$
$$R_m = \mu_m B Y_m.$$

The use of R rather than q is to distinguish between in situ reactions and outside injections. The terms are then used to determine the reaction term of the depletion of nutrients. That is,

$$R_n = -R_b - R_m.$$

While this does roughly describe the processes, some flaws have been identified [18]. The only time the system will be in equilibrium is after the nutrients are all consumed. Meaning that as long as there are nutrients available, the bacterial growth may be infinite. This could be rectified by adding a term that models a death rate of the microbes. However, again considering their robustness, that isn't a realistic option to bound their growth. Unboundedness is a problem mathematically, but, having bacterial growth limited by the access to a food source is a reasonable scenario. This trait is commonly referred to as a limiting nutrient. In an MEOR setting, at the very least there will be nutrients injected from wells and perhaps microbes as well. While this would lead to a constant food supply in the near-well region, further away from the injection well, the nutrients will have been depleted during transport resulting in a production equilibrium condition for the microbes. From a perspective of recreating results, it is logical to use the same equations used previously. The Monod model is used in both the works that will be recreated later. Therefore no guesswork must be done to determine the maximum growth rates or yield values. Finally, the advantages and disadvantages of using the empirical Monod model are well known and documented. There have even been attempts made to alter equation (4.1) to specifically describe bacterial growth in porous media [7]. For these reasons, we will proceed with the Monod model and combine it with our oil-water reservoir model. As the components of MEOR are microscopic particles, they are merely transported in the water phase and there is no explicit change to the velocity equation (3.8). Using the same method to introduce the formation volume factor β_i

and to modify injection from the outside q_i into $\tilde{q}_i$, the full system of conservation equations is

$$\begin{aligned}
\text{Oil:}\quad & \frac{\partial}{\partial t}\left(\frac{S_o\phi}{\beta_o}\right) + \nabla\cdot\left(\frac{\vec{u}_o}{\beta_o}\right) - \tilde{q}_o && = 0\\
\text{Water:}\quad & \frac{\partial}{\partial t}\left(\frac{S_w\phi}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w}{\beta_w}\right) - \tilde{q}_w && = 0\\
\text{Bacteria:}\quad & \frac{\partial}{\partial t}\left(\frac{S_w\phi B}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w B}{\beta_w}\right) - \tilde{q}_b - R_b && = 0\\
\text{Metabolites:}\quad & \frac{\partial}{\partial t}\left(\frac{S_w\phi M}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w M}{\beta_w}\right) - R_m && = 0\\
\text{Nutrients:}\quad & \frac{\partial}{\partial t}\left(\frac{S_w\phi N}{\beta_w}\right) + \nabla\cdot\left(\frac{\vec{u}_w N}{\beta_w}\right) - \tilde{q}_n - R_n && = 0
\end{aligned}$$

As the metabolites are solely produced inside the reservoir, there is no source term, only a reaction term, for the metabolite conservation. With the equations for transport in place, the model for metabolite effects must now be developed.

4.2 Metabolite Modeling

Different species of bacteria produce different metabolites which in turn have different effects on oil recovery. For this reason, they have been modeled generically up to this point. There are two types of metabolites that are of interest to this work, surfactants and polymers. Surfactants serve to reduce IFT, thereby combating the effects of capillary trapping. Polymers increase the viscosity of the water which leads to improved volumetric sweep of the water flood.

4.2.1 Biosurfactant

Capillary trapping was previously mentioned as one of the main reasons more oil is not recovered from reservoirs with only water flooding. This effect was then modeled through the introduction of relative permeability. It then stands to reason that to model the effect of surfactant, changes must be made to the relative permeability. To begin, we must understand how the surfactant works. As mentioned previously, surfactant molecules have both a hydrophilic and a hydrophobic end. This causes them to seek out the facial boundary between oil and water when in the reservoir. This can reduce the IFT three to four orders of magnitude which increases the mobility of the oil [17]. Not all surfactants are identical of course. There will be some variation in their effect on the IFT. A common trait is that a minimum concentration must be obtained before any change in IFT is seen, and that there exists a critical concentration level, past which there is no further effect on IFT. The following empirical equation serves to demonstrate the surfactant's effect on the IFT σ,

$$\sigma^*(M_s) = \sigma\frac{-\tanh(l_3\,M_s - l_2) + 1 + l_1}{-\tanh(-l_2) + 1 + l_1}. \tag{4.2}$$

The new IFT value σ^* is a product of the original IFT, the concentration of surfactant, M_s, and various properties of the surfactant, l_j [17]. Varying the values of l_j will determine how effective the surfactant is and some different examples may be seen in Figure 4.1. The values for l_i and σ were taken from [17] and correspond to different minimum and critical concentration values as well as different curvatures. The most effective surfactant is drawn in blue and that is the example which we will later attempt to recreate. We now seek to alter the relative permeability equations (3.6) to account for

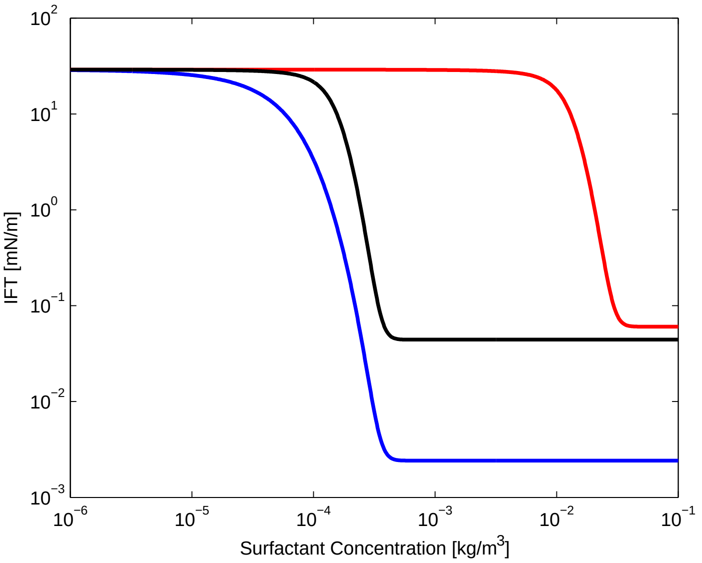


Figure 4.1: The graph of three different surfactants' effect on IFT is shown. They show differences in curvature, minimum concentration, and critical concentration values.

the change in IFT. There are a few methods that already attempt to do this and take the IFT into explicit consideration. One method uses the capillary number,

$$N_{ca} = \frac{\eta_w u}{\sigma^*}.$$

It has been determined that a lower capillary number corresponds to a lower residual oil value [16]. Thus, a new value for S_{or} is found based on the capillary number and entered into equation (3.6) to obtain a better estimate of the relative permeability. Another method is Coats' correlation. This method uses the following relation,

$$f(\sigma*) = \left(\frac{\sigma^*}{\sigma}\right)^{\frac{1}{a}} \tag{4.3}$$

where the exponent a typically ranges between 4 and 10 [16]. New residual saturation values are found as,

$$\begin{aligned} S_{wi}^* &= f(\sigma^*)S_{wi} \\ S_{or}^* &= f(\sigma^*)S_{or}. \end{aligned} \tag{4.4}$$

These values are similarly used in equation (3.6) to determine new relative permeability curves, $kr_{i,base}$. Another curve, $kr_{i,misc}$ is found as a straight line between the endpoints of $kr_{i,base}$. This simulates the behavior of fully miscable fluids. The following relation is then used to determine the ultimate relative permeability value,

$$kr_i^* = f(\sigma^*)kr_{i,base} + (1 - f(\sigma^*))kr_{i,misc}.$$

Using the same example as before with $S_{wi} = 0.16$ and $S_{or} = 0.23$, this method is applied to the curves in Figure 3.1 with a surfactant that reduces the IFT three orders of magnitude. The results may be seen in Figure 4.2. The clearest difference is that the new residual saturation values are $S_{or}^* = 0.09$ and $S_{wi}^* = 0.06$. However, using Coats' interpolation method, there is no increase in the maximum relative

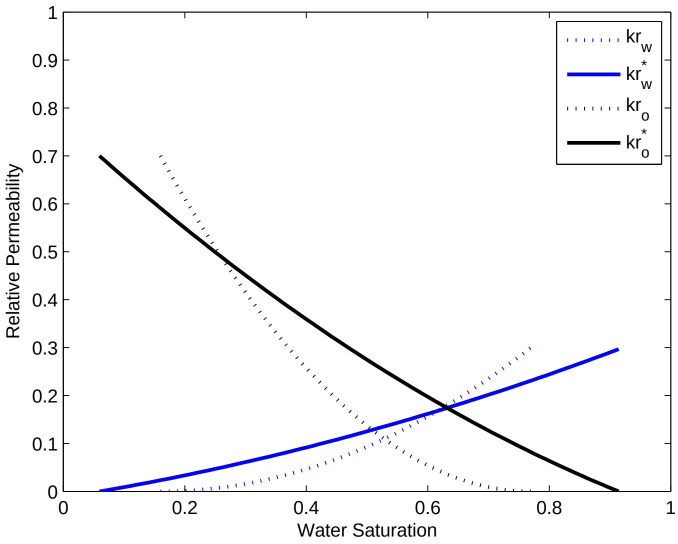


Figure 4.2: Coats' interpolation method. After new values are found for S_{wi} and S_{or}, two new relative permeability curves are found for each phase. The new relative permeability curve depends on the current IFT value.

permeability. This means that there are saturation levels where the relative permeability is less than what it would be without the use of biosurfactants. This is not the case for another method presented in Nielsen et al. (2010) [16] where all the parameters of equation (3.6) are altered. Continuing the use of equations (4.3) and (4.4), new values are found for kr_{owi}, kr_{wor}, and the Corey exponent, n, using the following formula

$$\chi^* = f(\sigma)\chi + (1 - f(\sigma)). \tag{4.5}$$

In this form, χ takes the place of whatever parameter is being updated. With all the new parameters due to the biosurfactant concentration, the relative permeability is calculated again with equation (3.6). Using the same example as before, we find our new curves and they are shown in Figure 4.3. Now the relative permeability values at residual saturation are increased. This method succeeds in improving the relative permeability at every saturation level as opposed to Coats' method. For that reason, this is the method that will be used to model the effect of biosurfactant in our model.

Another consideration to be made is how the surfactant behaves at the boundary of the two phases. It is reasonable to consider that some of the surfactant is transported into the oil phase where it will no longer effect the IFT [17]. This leaves only the portion of surfactant in the water phase available to alter the IFT and thereby the relative permeability. To reconcile this with our model, partitioning of the surfactant between the phases is done. The method will be the same as is used in Nielsen et al. (2010) [16]. The ratio of surfactant in water to surfactant in oil is proportional to the masses of the two phases,

$$\frac{M_{sw}}{M_{so}} = \mathcal{K}\frac{S_w\rho_w}{S_o\rho_o}. \tag{4.6}$$

The concentrations of surfactant in water and oil of course add up to the total surfactant concentration,

$$M_{sw} + M_{so} = M_s.$$

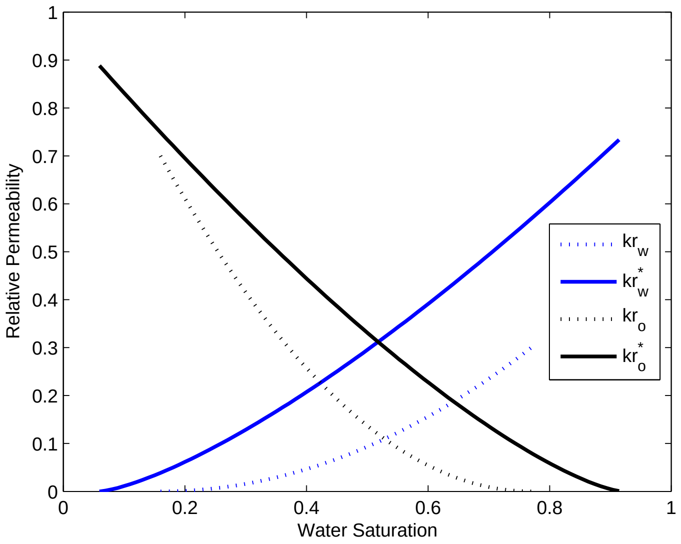


Figure 4.3: Corey interpolation method. Here it can be seen that not only are the residual saturation levels reduced, but also the maximum relative permeability values.

Therefore, to determine the amount of surfactant in the water phase, a substitution of $M_{so} = M_s - M_{sw}$ is made to equation (4.6),

$$\begin{aligned}
\frac{M_{sw}}{M_s - M_{sw}} &= \mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} \\
M_{sw} &= M_s\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} - M_{sw}\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} \\
M_{sw} + M_{sw}\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} &= M_s\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} \\
M_{sw}\left(1 + \mathcal{K}\frac{S_w\rho_w}{S_o\rho_o}\right) &= M_s\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o} \\
M_{sw} &= \frac{M_s\mathcal{K}\frac{S_w\rho_w}{S_o\rho_o}}{1 + \mathcal{K}\frac{S_w\rho_w}{S_o\rho_o}}
\end{aligned}$$

With the partitioning coefficient $\mathcal{K}$ being the only value to be defined, we now have a method to determine the amount of surfactant influencing the IFT. When coupled with the Corey parameter interpolation method, this constitutes a full surfactant model.

4.2.2 Biopolymer

The other type of metabolite we are interested in is biopolymer. It is produced in the same way as the biosurfactant and will be denoted similarly as M_p in units of kg/m^3. The effect of the polymer is an increase in water viscosity, η_w. As can be seen in equation (3.8), the direct effect is that the velocity of the fluid is slowed. This reduces the mobility of the water phase, making it more similar to the mobility of the oil. The displacement is more effective in this manner and the sweep of the water flood increases. The problem of the water flood channeling through a thief zone of high permeability is greatly reduced with polymer EOR. As the viscosities of the water and polymer are typically known,

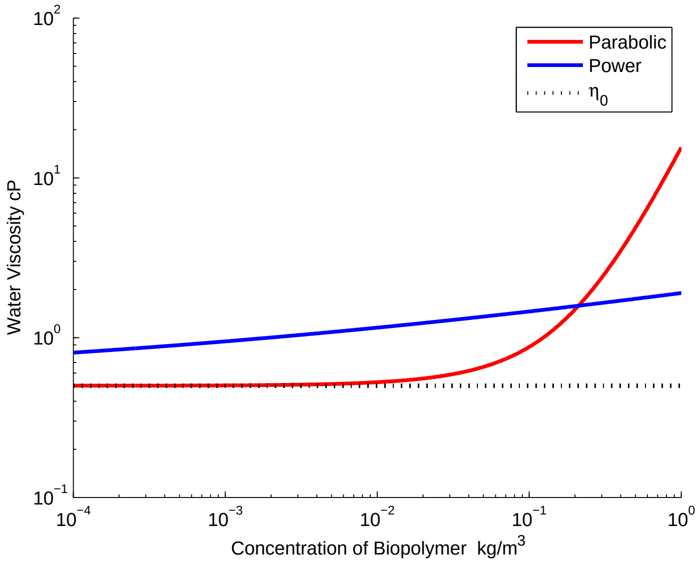


Figure 4.4: A plot of the viscosity effects modeled by equations (4.7) in red and (4.8) in blue. The initial viscosity value is taken at 0.5 cP.

the viscosity of a mixed solution can be found through experimentation and interpolation. The effects of biopolymer are simulated in Lacerda et al. (2012) [7] using three different equations. The first uses a linear relationship between the original water viscosity and the concentration of biopolymer,

$$\eta_w^* = \eta_w + GM_p$$

where G is a constant. This is by far the simplest method which typically means that while it may be easy to implement, it does not guarantee a high degree of realism. The next method presented is a parabolic relationship for changing the viscosity,

$$\eta_w^* = 0.414M_p^2 + 1.895M_p + 0.071.$$

The most glaring aspect of this equation is that the original viscosity is not present. Lacerda et al. cite the source of this equation as stemming from a paper written by Bae et al. (2008) [2], however, after some consideration, we do not understand how this was extracted. In its place, we will use another parabolic equation found in an article by Bartelds et al. (1997) [3], also modeling the effects of polymer flooding. This equation does not simply add on to the viscosity, but instead increases it by a factor,

$$\eta_w^* = \eta_w((5M_p)^2 + 5M_p + 1). \tag{4.7}$$

The final method presented in Lacerda et al. for describing the biopolymer viscosity effect is referred to as the power law and has the following form,

$$\eta_w^* = \eta_w + 1.4019M_p^{0.1653}. \tag{4.8}$$

As the linear relationship requires a constant to be defined and this is not given, we will ignore it and move forward with equations (4.7) and (4.8) and compare and contrast them. A graphical comparison of their behavior can be seen in Figure 4.4 for different concentrations of biopolymer. The initial viscosity value is taken as $\eta_w = 0.5$ cP. The figure shows that there is initially very little change when using the parabolic relationship until about 0.02 kg/m^3 concentration of biopolymer is reached. Then it rapidly increases and overtakes the power law curve at about 0.1 kg/m^3 and continues to increase. Conversely, the power law curve displays a steady slow increase in this interval though has nearly doubled the viscosity at only 0.0001 kg/m^3 of biopolymer. The problem with both of these equations is that they only simulate a single, not necessarily realistic, polymer. In order to increase realism, and hence the value of the model, we must develop the model further.

As polymer injection is a widely used EOR technique, there already exists a model in MRST to simulate this which will now be presented. Not only can different viscosity changes be simulated, but it also takes into account the time to mix the polymer, and adsorption of the polymer onto the pore walls. As the computational cells may be quite large, it is a gross simplification to assume that the polymer instantly diffuses across the entire volume. To increase the realism, it is common to use the Todd-Longstaff model of mixing [21]. This model assumes that the viscosity of a fully mixed polymer solution, η_m, is a known function of the concentration, M_p, and that this reaches a maximum value, $\eta_p = \eta_m(M_{p,\mathrm{max}})$. The effective viscosity of the polymer is then

$$\eta_p^* = \eta_m(M_p)^\omega \eta_p^{1-\omega}$$

where $\omega \in [0,1]$ is a mixing parameter. With $\omega = 0$, there is no mixing and conversely, with $\omega = 1$ complete mixing is obtained. A ratio that will be used often in this model, $M_p/M_{p,\mathrm{max}}$ will henceforth be notated $\bar{M}_p$. As the polymer and water may partially mix, decreasing the polymer viscosity while increasing the water viscosity, it is useful to define the partially mixed water viscosity as

$$\eta_{w,\mathrm{mix}} = \eta_m(M_p)^\omega \eta_w^{1-\omega}.$$

The effective water viscosity is then determined by the following relation

$$\frac{1}{\eta_w^*} = \frac{1-\bar{M}_p}{\eta_{w,\mathrm{mix}}} + \frac{\bar{M}_p}{\eta_p^*}.$$

The new viscosity values η_w^* and η_p^* are then used in the Darcy equation (3.2) to find the velocities $\vec{u}_w$ and $\vec{u}_m$ respectively. It is also possible to determine the polymer mixture velocity using the following relation

$$\begin{aligned}\vec{u}_m &= \frac{\eta_{w,\mathrm{mix}}}{\eta_p^*}\vec{u}_w \\ &= \left[\left(1-\bar{M}_p\right)\left(\frac{\eta_p}{\eta_w}\right)^{1-\omega} + \bar{M}_p\right]^{-1} \vec{u}_w.\end{aligned}$$

If this was simply an oil-water-polymer model, it would make sense to have the velocity in the polymer conservation equation replaced by $\vec{u}_m$ to account for mixing. This allows for phases with different properties to flow accordingly. The problem in our MEOR setting is that we would then have to consider the concentration of nutrients and bacteria that are being transported at different velocities. However, one of the assumptions made with MEOR modeling and simulations is that the bacteria and nutrients are fully dispersed in each computational volume. It then stands to reason that the biopolymer metabolites would be similarly dispersed when they are produced. Mixing is introduced in polymer EOR models because the polymer is injected into the reservoir in highly concentrated quantities and logically this does not disperse instantaneously in water. The polymer possesses a different density and much higher viscosity than the water and will naturally flow differently. In MEOR, only nutrients and bacteria are injected and while the nutrients may be a viscous corn syrup or similar, dispersion should be much faster than for polymer injection. If mixing were to be considered in our model, we would need to include it for all transported substances. We assume that the dispersion time of nutrients and bacteria is negligible and therefore the Todd-Longstaff mixing model is not needed.

Another common aspect of a polymer EOR model is adsorption. The adsorption is assumed to occur instantly and follow a known function of the polymer concentration. The adsorbed polymer will be denoted $M_{p,a}$. The adsorption is important not only because it stops some polymer from flowing, but also since it can decrease the permeability in the reservoir. This is modeled by a function $R(M_{p,a})$ that is inserted into the Darcy equation,

$$\vec{u}_p = -\frac{kr_w}{\eta_p^* R(M_{p,a})}\mathbf{K}(\nabla p - \rho_w g \vec{e}_z).$$

The conservation equation for the polymer is then,

$$\frac{\partial}{\partial t}\left(\frac{S_w \phi M_p}{\beta_w}\right) + \frac{\partial}{\partial t}\left(\rho_{\text{rock}}(1-\phi)M_{p,a}\right) + \nabla \cdot \left(\frac{\vec{u}_m M_p}{\beta_w}\right) - R_m = 0.$$

When including adsorption into the biopolymer model, it is of course possible to set $M_{p,a} = 0$ to avoid modeling adsorption. Then the only change is made to the viscosity by

$$\eta_w^* = \eta_m(M_p).$$

In this case we may just use the parabolic or power laws to model the viscosity as before. While this polymer model is more realistic now with the addition of adsorption, it only investigates how the flow is retarded and assumes that the porosity does not decrease enough to make a difference to the pore volume. Additionally, it is of course also possible for adsorption to occur with the the other transported substances as well. It could be considered inconsistent to include adsorption just for polymer. However, studies have shown that adsorption occurs relatively rarely for nutrients and surfactant compared to bacteria [17]. Therefore we will retain adsorption for polymer and assume that the effects of adsorption for surfactant and nutrients are negligible. Bacteria adsorption is actually much more complicated and will require additional work.

4.3 Biofilm Modeling

When bacteria adsorb to the rock surface in the reservoir, overall bacterial transport is considerably hindered. This actually results in more metabolites being produced closer to the injection sites. Additionally, pore clogging may occur. This has been modeled in several ways. The bacteria may form a biofilm that coats the pore walls which is either modeled as a homogeneous or heterogeneous structure [20]. It is also possible that the bacteria form colonies while suspended in the water phase that eventually become too large to pass through any pores [11]. We will only consider homogeneous biofilm as complicated microscopic structures are not reconcilable on our macroscopic scale. Therefore we wish to model bacteria adsorption and biofilm formation with both a change in porosity and permeability. We assume that any biopolymer adsorption will not hinder biofilm formation. This is defensible because in truth biofilm consists of more than bacteria cells but also extracellular polymeric substances (EPS) that could amalgamate with the biopolymer. The EPS is effectively the cause of bioclogging as it can constitute 95% of the biofilm [20]. The first consideration that will be made is the addition of a new conservation equation for biofilm. Because the bacteria are not rendered inactive after adsorption, they still consume nutrients, produce metabolites, and reproduce, they must be thought of simply as bacteria that are not transported. It will be assumed that any new bacteria produced by the adsorbed bacteria will not automatically be added to the biofilm. Also, although adsorption is by no means irreversible, degradation of biofilm will not be considered. Therefore, the conservation equation for biofilm will not have a flux term. We will differentiate between free-floating bacteria and adsorbed bacteria with B_f and B_a respectively where the total volumetric concentration is still B. We have B_a in units of kg/m^3 pore volume. The free-floating bacteria conservation equation is then,

$$\frac{\partial}{\partial t}\left(\frac{S_w \phi B_f}{\beta_w}\right) + \nabla \cdot \left(\frac{\vec{u}_w B_f}{\beta_w}\right) - \tilde{q}_b - R_b = 0.$$

The reaction term R_b has also changed now and we will specify exactly how later. First we must define how the bacteria adsorption itself will be modeled. We choose the Langmuir model with equilibrium adsorption for this purpose. This means that the adsorption kinetics are ignored and a partitioning of bacteria between the water phase and biofilm occurs immediately. This retards the transport of bacteria which then results in nutrients being transported with no bacteria to consume them [22]. Nevertheless, this will serve adequately for the purpose of modeling adsorption in our model. The equation takes the form,

$$\frac{B_f}{\mathcal{L}} = \frac{B_f}{\omega_2} + \frac{1}{\omega_1 \omega_2},$$

where ω_1 and ω_2 are constants that determine the rate of adsorption. ω_1 is in units of kg/m^2 and represents the maximum attainable level of biofilm. The constant ω_2 determines the speed of adsorption and is the inverse of the so called affinity constant with units m^3/kg [23]. We use $\mathcal{L}$ to represent the mass of bacteria adsorbed per unit area kg/m^2. The equation may be altered to obtain,

$$\mathcal{L} = \frac{\omega_1 \omega_2 B_f}{1 + \omega_2 B_f}, \tag{4.9}$$

resulting in the concentration of adsorbed bacteria being in units kg/m^2. In some work, this may be the only form of the equation presented and $\omega_1\omega_2$ in the numerator may just be defined as ω_1. The idea of this equation is that the concentration of adsorbed bacteria $\mathcal{L}$, will be used to calculate the mass of bacteria per unit pore volume B_a. To model the amount of biofilm, there are some physical considerations to be made. Firstly, biofilm may only exist in the water phase since the bacteria are being transported in water and no partitioning between phases occurs. Also, as the bacteria adsorb to the pore walls, the porosity and subsequent surface area available must be accounted for. The typical surface area of porous rock, A_v, ranges between $10^5 - 10^6$ m^2/m^3 volume [17]. Thus, the surface area available for adsorption is,

$$\mathcal{S} = \frac{S_w}{\phi} A_v.$$

Thus multiplying this by attached bacteria concentration we obtain,

$$B_a = \mathcal{S}\mathcal{L}.$$

We now can use this to find the biofilm partition which is traditionally denoted as σ. However, since we are already using this Greek letter for the IFT, we instead use ψ. This is determined by using the density of the biofilm, ρ_b, which can range around 1000-1300 kg/m^3 [11, 17]. Then,

$$\psi = \frac{B_a}{\rho_b},$$

represents the amount of pore space now clogged such that the updated porosity is,

$$\phi^* = \phi_0(1 - \psi),$$

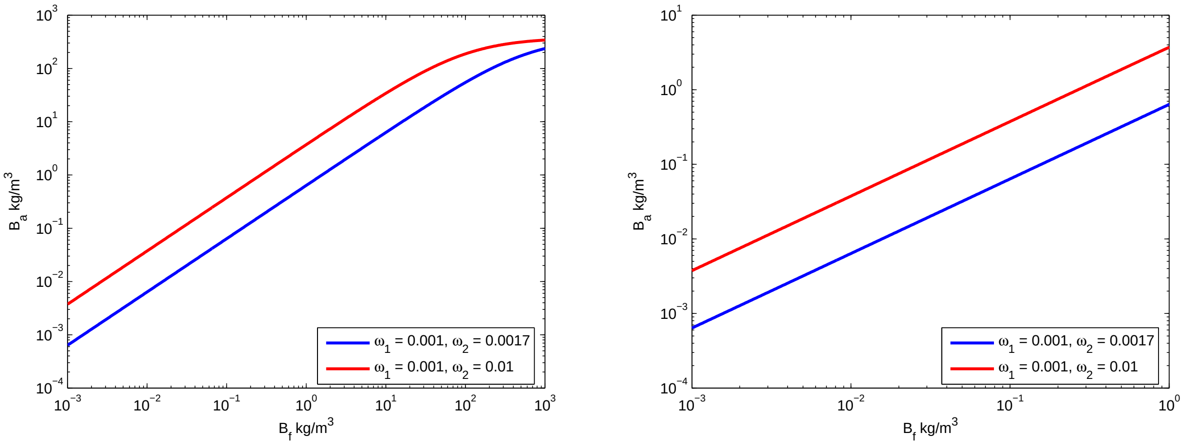


Figure 4.5: Solutions of equation (4.9) with different ω_2 values. The graph on the left displays the asymptotic behavior while the graph on the right has concentration values in our expected range.

where ϕ_0 is the initial porosity value. From here there are several different methods forward to alter the absolute or relative permeabilities. A comparison of 10 different alterations of permeability is presented in an article by Thullner et al. (2010) [20]. Here the equations are altered according to the relative change in porosity,

$$\phi_{\text{rel}} = \frac{\phi^*}{\phi_0} = 1 - \psi.$$

Their conclusion was that bioclogging in controlled experiments is hard to predict and it is consequently much more difficult on the field scale. Even the most complex alterations are imperfect. For that reason, we choose a simple path that alters the water relative permeability,

$$kr_w^* = kr_w(1-\psi)^{19/6}.$$

This relation was developed to describe the macroscopic effects of only the formation of biofilm reducing the pore radius and did not consider pore size distribution or complete clogging of pores. Therefore, it is congruent to the scope of our model as well. To improve understanding of how the adsorption and biofilm model work, Figure 4.5 displays two different solutions to equation (4.9) with different parameters. In this example, $\omega_1 = 0.001$ corresponds to a maximum of 75% pore volume being occupied. This is done with $A_v = 3 \times 10^5$, $S_w = 0.5$, $\phi = 0.4$, and $\rho_b = 1000$. Including the biofilm density allows us also to visualize how the porosity is modified in Figure 4.6. From the figure, it can be seen that biofilm will not have a drastic effect on the porosity when the bacteria concentrations are as low as expected.

What remains is to keep the other MEOR processes consistent. This requires that metabolite production, bacterial growth, and nutrient consumption occur in the same manner as before. The reaction terms from the Monod model are updated to include biofilm,

$$\begin{aligned}
R_b &= \mu_b(B_f + B_a)Y_b \\
R_m &= \mu_m(B_f + B_a)Y_m \\
R_n &= -\mu_b(B_f + B_a)Y_b - \mu_m(B_f + B_a)Y_m.
\end{aligned}$$

The growth rates depend only on the limiting nutrient concentration and therefore remain unchanged. Now the biofilm model will function properly with the other aspects of the MEOR model and we may begin to consider simulations.

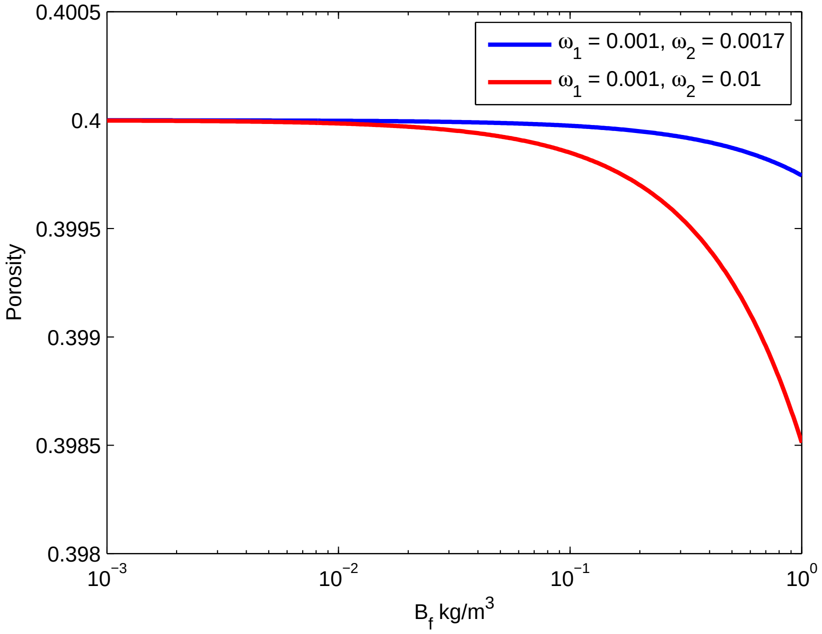


Figure 4.6: The change in porosity due to biofilm formation.

4.4 Assumptions

This section provides an overview of all assumptions used in this MEOR model.

1. The reservoirs are not inhospitable to bacteria.
2. Bacterial growth and metabolite production can be modeled by the Monod equation.
3. There is no indigenous bacteria present competing for nutrients.
4. The growth is only influenced by the limiting nutrient presence.
5. Chemotaxis, the ability of bacteria to influence their movement, does not occur.
6. Bacterial decay or deactivation is covered by an effective growth rate.
7. The Langmuir model of equilibrium adsorption is valid.
8. Only one metabolite is produced in sufficient quantity at a time. Any others produced are negligible.
9. Nutrient and biosurfactant adsorption are negligible.
10. Bacterial growth and production rates are identical in biofilm and in the water phase. Access to nutrients is not a problem for biofilm.
11. Biosurfactant can partition between the oil and water phase. This happens instantly. Biosurfactant in the oil phase has no effect.
12. Dispersion of bacteria and nutrient in the water phase is complete and immediate.

13. New bacteria and metabolites produced by biofilm are introduced into the water phase and not retained in the biofilm.

14. Biofilm forms on the pore walls decreasing the pore radius.

15. There is no change in the density of fluids due to the presence of bacteria, nutrients, or metabolites.

Chapter 5

MATLAB Reservoir Simulation Toolbox

SINTEF has been developing MRST as an open source toolbox for the purposes of rapid prototyping new models and computational methods for reservoir simulations in MATLAB. This is an ongoing project that is continually being improved upon by SINTEF Applied Mathematics in Oslo, Norway. The following is a brief overview of the basic functionality of MRST and more information can be found in [12]. The main idea of MRST is to create computational simulations of oil reservoirs that can emulate the expected behavior of the real world with the help of finite volume discretization. In order to achieve that goal, the first step taken is to define a reservoir structure. This can simply be a box or, if more realistic models are desired, an input file can be read defining the geometry of a complex simulation reservoir. For the purposes of the simulation, the reservoirs are segmented and represented by several control volumes referred to as cells. These volumes do not need to be cubic or even representable with a Cartesian coordinate system, though for simplicity, that is all that will be considered here. These cells are represented as data structures, each containing values to represent the size, location, pressure, porosity, absolute permeability, phase saturation, and a reference to the bordering cells. Together they form the computational version of the reservoir. To simulate oil recovery, injection and production wells must be simulated as well. These may be defined anywhere in the reservoir as either desired rates or pressures. Alternatively, source points and boundary conditions may also be implemented. No matter the choice, the cells will then be modified with source or sink functionality for simulation. The fluid properties must also be defined, namely, density, viscosity, and the relative permeability to be simulated. The reservoir must also be given an initial state from which to begin simulations. As with the geometry, simulations can be either simple or complex. On the conceptually simpler end of the spectrum is defining a constant flux boundary condition at one wall of the reservoir. Then specifying the length of the simulations and the desired number of steps to be generated as output. On the other side, a more complex simulation method could involve several different injection and production wells using different injection rates and even varying these rates during the simulation. It is not uncommon for academic simulations to be done in a box structure with one side of the reservoir being given a constant injection flux rate [7, 16]. Figure 5.1 shows two possible examples of reservoirs in MRST. Both are shown with their pressure gradients visualized, high pressure represented by red and comparatively low pressure by blue. The simple reservoir with an injection flux boundary condition on the red side and a constant pressure boundary condition on the blue side. The reservoir on the right is part of the Sensitivity Analysis of the Impact of Geological Uncertainties on Production (SAIGUP) project [5]. It is a synthetic reservoir with complex variations in porosity and permeability similar to what is found in real reservoirs. The wells are placed randomly

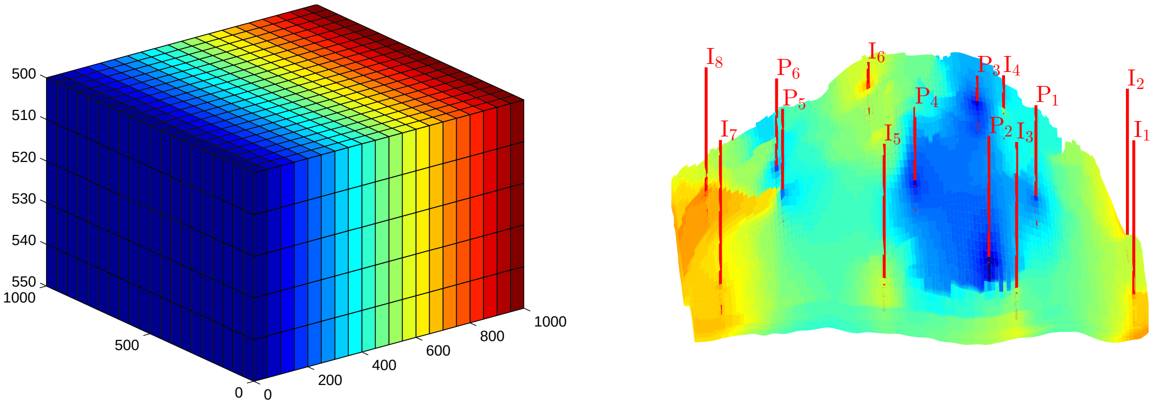


Figure 5.1: Two examples of reservoirs are seen. On the left, a simple box structure using boundary conditions. On the right, a synthetic, though complex, SAIGUP reservoir with several injection and production wells, marked I and P respectively.

and injection wells are marked with 'I' while production wells are marked with 'P'. It can be seen that the pressure is lower around the production wells and higher around the injection wells. Though distorted in the figure, the SAIGUP reservoir is both shallow and long in comparison to its width.

5.1 Simulation

With the reservoir constructed and all the rock, fluid, and well properties defined, the stage is set to begin simulations. This is done by defining what is referred to as the schedule. Aptly named, it controls the length of time the reservoir should be simulated for and also the intermediate steps at which to produce output. It is here that the strength of MRST is revealed. When the model is discretized, there are multiple variables being updated at each time step. For a two-phase system, this includes saturation and pressure values. If a production well is present, the flux through that well must also be solved for. Using the wells as source terms and the transfer of fluids from one cell to another as fluxes, we can then observe the cells with respect to equations (3.7) and (3.8),

$$\frac{\partial}{\partial t}(S_i \phi \rho_i) + \nabla \cdot (\rho_i \vec{u}_i) - q_i = 0, \qquad \vec{u}_i = -\frac{kr_i \mathbf{K}}{\eta_i}(\nabla p - g\rho_i \nabla z).$$

The first equation is the mass conservation and the second is Darcy's Law. The equations are discretized using a first-order implicit scheme,

$$\begin{aligned} \frac{(S_i\phi\rho_i)^{n+1} - (S_i\phi\rho_i)^n}{\Delta t^n} + \textsc{div}(\rho_i \vec{u}_i)^{n+1} - q_i^{n+1} &= 0 \\ \vec{u}_i^{n+1} = -\frac{kr_i \mathbf{K}}{\eta_i^{n+1}}(\textsc{grad}(p^{n+1}) - g\rho_i^{n+1}\textsc{grad}(z)), \end{aligned}$$

where GRAD and DIV are discrete gradient and divergence operators respectively. The superscript of n and $n+1$ are to indicate new and old values where Δt is the length of the time step. The discretization also accounts for compressibility considerations which is why ρ, ϕ,and η may change at each iteration. This is the modeled in MRST with a nonlinear dependence on the pressure, p. The operator DIV calculates the fluid flux between each of the boundaries of the cell into its neighboring cells. When dealing with cubic cells, this means that there are 6 values to combine, one for each border of the cell. The direction of fluid flow must of course be taken into account in order to arrive at the net flux of the cell since fluid will typically be both leaving and entering the cell. Whereas the DIV operator is a mapping from the cell boundaries to the cell, the GRAD operator is a mapping from the cell to the boundaries. In fact, it is the negative adjoint of the divergence operator as

$$\int_\Omega p \nabla \cdot \vec{u}\, d\Omega = -\int_\Omega \vec{u} \cdot \nabla p\, d\Omega.$$

The proof that this holds in the discrete setting can be found in the MRST guidebook [12]. These operators are then used to evaluate the equations for velocity and conservation. In a two-phase oil water system, this results in conservation equations for each fluid. After the reservoir simulation equations have been defined, the system must be solved. Solutions are found simultaneously for all variables in all grid cells at each time step. This is accomplished in MRST by using Newton's method.

5.2 Newton's Method

The Newton method, also referred to as Newton-Raphson, is a conceptually straight-forward approach to solving a system of the form

$$f(x) = 0.$$

Given some initial guess to the solution, x_0, the method takes a step to improve upon the first estimate

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

where f' denotes the derivative of the function f. This is repeated n times to find

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

such that

$$f(x_{n+1}) = 0.$$

Obviously in a numerical setting, some tolerance is used when evaluating the accuracy of the solution. This method is not guaranteed to converge. It is possible for the iterations to cycle through the same values, or for $f'(x_n) \approx 0$ resulting in non-finite values for x_{n+1}. In a multi-variable setting, the problem then becomes

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

where $\mathbf{F}$ is now an operator and $\mathbf{x}$ is a vector of variables. The concept remains the same but the solution method is altered to avoid computationally expensive matrix division. This results in

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n - \mathbf{F}(\mathbf{x}_n)(D\mathbf{F}(\mathbf{x}_n))^{-1} \\ \mathbf{x}_{n+1} - \mathbf{x}_n &= -\mathbf{F}(\mathbf{x}_n)(D\mathbf{F}(\mathbf{x}_n))^{-1} \\ D\mathbf{F}(\mathbf{x}_n)(\mathbf{x}_{n+1} - \mathbf{x}_n) &= -\mathbf{F}(\mathbf{x}_n), \end{aligned}$$

where $D\mathbf{F}$ is the Jacobian of $\mathbf{F}$. The unknown value becomes

$$\delta\mathbf{x}_{n+1} = \mathbf{x}_{n+1} - \mathbf{x}_n$$

and the problem is now a linear system which can be solved efficiently. The difficulty posed by this problem is finding the Jacobian matrix $D\mathbf{F}$. In a smaller system, this could be found analytically, however this can be quite time consuming and must be recalculated when there is any alteration to **F**. This is applicable to MRST because the conservation equations constitute **F** while the variables form **x**. As MRST is designed for rapid prototyping of new methods and models, tedious Jacobian determinations present a hindrance in achieving that goal. In order to reconcile this, MRST uses automatic differentiation techniques, therefore any changes to the equations does not affect the solution procedure.

5.3 Automatic Differentiation

The power of the solvers in MRST reside in its implementation of variables that can be automatically differentiated. This replaces the time consuming calculations of Jacobi matrices that are necessary when solving the equations to simulate the reservoir. Automatic differentiation evaluates the matrices arithmetically from the given variables in the equations. When an equation is created using these variables, the Jacobi matrix is automatically formed. For a system of 2 equations with 2 variables, it is not hard to determine the Jacobian. For example, given the equations

$$\begin{aligned} f_1 &= x_1^5 + 3x_1^2x_2^3 + 9x_2 \\ f_2 &= 4x_2^3 + x_1, \end{aligned} \tag{5.1}$$

the corresponding Jacobi matrix is

$$J = \begin{bmatrix} 5x_1^4 + 6x_1x_2^3 & 9x_1^2x_2^2 + 9 \\ 1 & 12x_2^2 \end{bmatrix}. \tag{5.2}$$

Now however, when solving for two-phase flow in MRST, there are six variables with multiple equations for each grid cell. This means there are thousands of elements in the Jacobi matrix that must be evaluated thousands of times in the course of a single simulation. Automatic differentiation, as its name implies, accomplishes this automatically. MRST allows the initiation of variables that keep track of the derivative. In other words, if a variable x is initiated as an automatically differentiable variable, any calculations $f(x)$ would automatically compute $\frac{d}{dx}f(x)$. Continuing with (5.1) and (5.2), imagine

$$\begin{aligned} x_1 &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix} \\ x_2 &= \begin{bmatrix} 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \end{bmatrix}. \end{aligned}$$

To evaluate the Jacobian at each of the paired points (x_{1i}, x_{2i}), where i is the index from 1 to 10, the variables must simply be initiated and the equations formed. When the equations are combined, a 20 × 20 sparse matrix is formed automatically containing the Jacobi matrix values in block form,

$$J = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} \end{bmatrix}.$$

While this is still not an overly complicated calculation, when thousands need to be made with increasingly complex equations, automatic differentiation becomes an invaluable asset. Its implementation in MRST allows for simplified simulations and the ability to add and alter equations without worrying about the Jacobian.

5.4 Implementation of MEOR Model

From a computer science standpoint, an advantage of MRST is its use of object oriented programming. The code has been created in such a way that there are several different layers of classes with

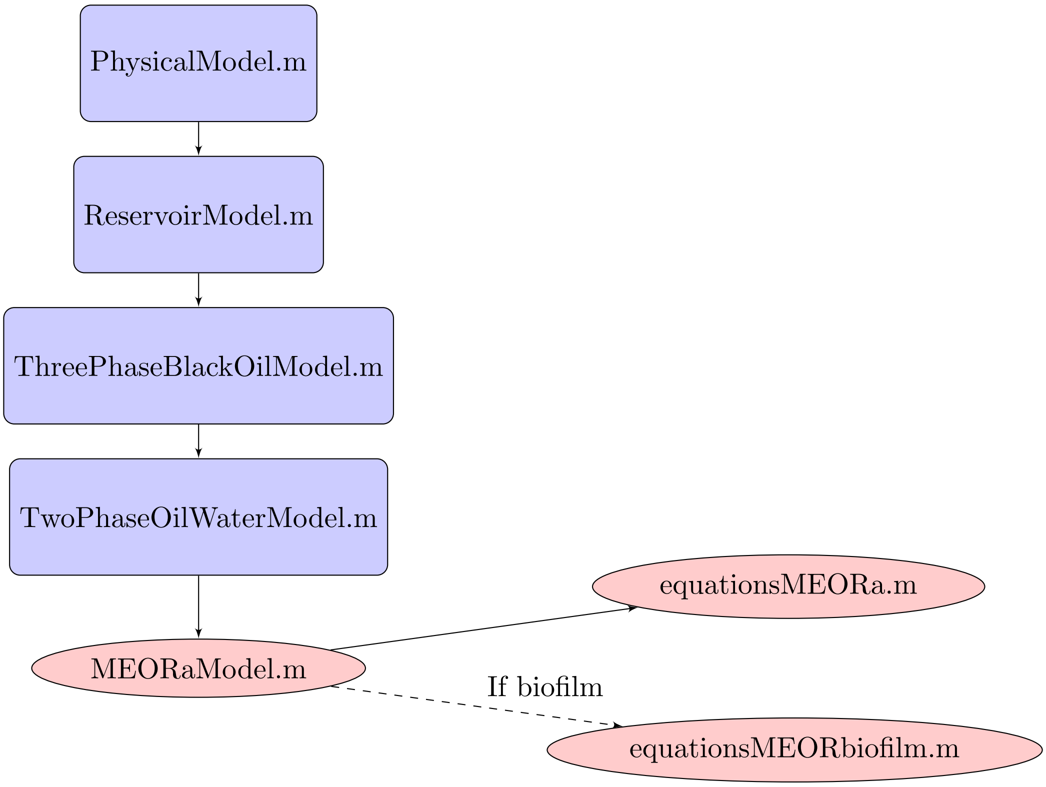


Figure 5.2: Diagram of MRST model inheritance. The new MEOR specific model and equations are in red.

inheritance. This is done to minimize code repetition and simplify the process of adding a different type of model. All reservoirs must be created and solved for using the same process so it is intuitive to write the code this way to simplify the process and reduce the possibility of mistakes. As this is the case, the addition of the MEOR model requires mainly a new type of model structure and a new equation file. Several other small changes are necessary to handle new keywords as well. The solution procedure, computational geometry formation, processing of input files, and the handling of wells, boundary conditions, and source points is universal. This object oriented structure and automatic differentiation is how MRST succeeds as a tool for rapid prototyping. Figure 5.2 shows the parent classes of our new model and the child functions for the equations. The different models are data structures to keep track of the physical traits that are present in the different type of reservoirs. The super class PhysicalModel implements physical models for use with automatic differentiation. ReservoirModel accommodates the rock and fluid features and common variables and phases. TwoPhaseOilWaterModel is a child of ThreePhaseBlackOilModel because the latter allows for dissolved gas and vaporized oil whereas the former is essentially a special case where those are not present. Finally, as our model has only considered oil and water phases, MEORaModel is the final subclass in the hierarchy. The MEOR model is a data structure that keeps track of the new additions, specifically, the microbes, nutrients, metabolites, and biofilm. The specific type of metabolite, biosurfactant or biopolymer, is also an important attribute. The MEOR equation file uses the Monod equations to set up the microbial kinetics. Also inside the equation file is the effect of whichever metabolite is present in the reservoir. Biosurfactant is first partitioned according to equation (4.6), then the new values for the Corey parameters are found using equation (4.5). This is then all used to find the new relative permeability according to the Corey equation (3.6). If instead biopolymer is present, the model is changed

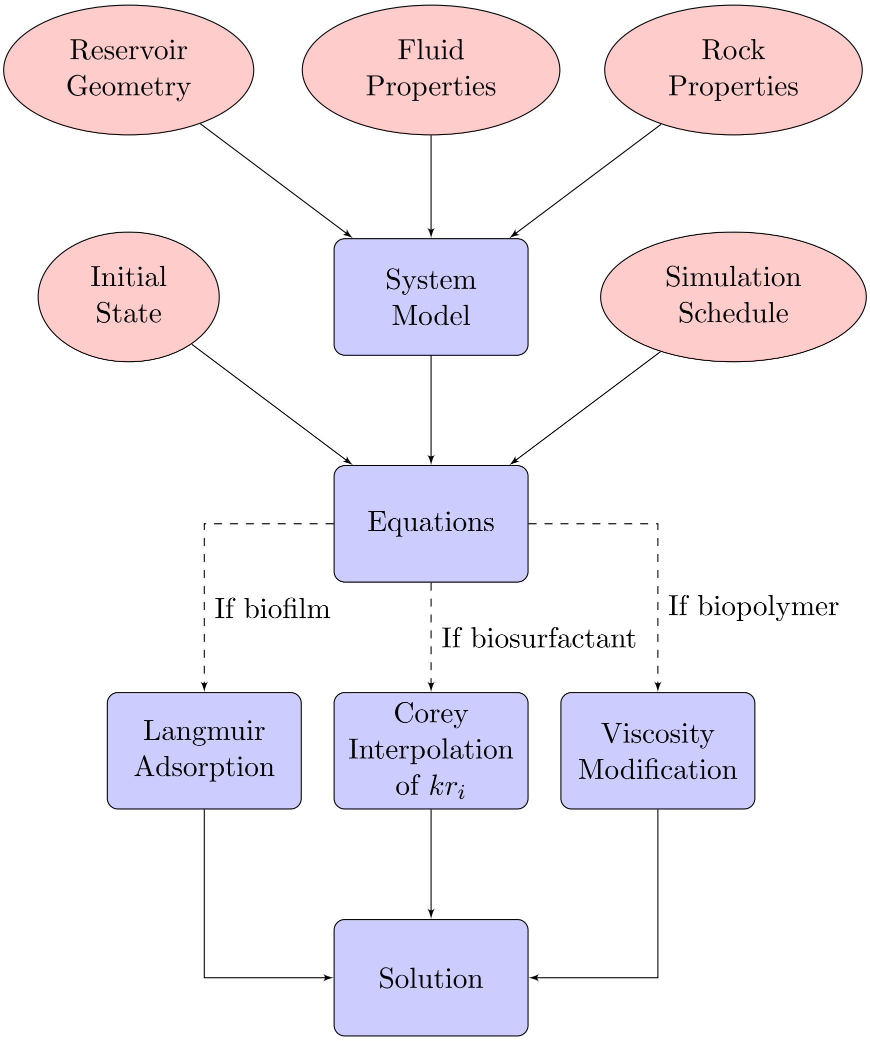


Figure 5.3: Flowchart presenting an overview of the solution procedure in MRST of the MEOR model. User input is presented in red.

to account for the change in viscosity, $\eta^* = \eta_m(M_p)$. If adsorption is desired, the relationship must be given by a paired list of concentrations and their corresponding adsorption values. A function is then found by interpolating values between the pairs. The same method may also be used for the viscosity change. Biofilm is treated as a special case since we add a new conservation equation. The effects of bioclogging, changes to the absolute permeability and porosity of the reservoir, are not implemented for a pair of reasons. The first is that bioclogging effects do not typically make a marked difference on recovery unless the injection concentration of nutrients is unrealistically high [17]. The second is simply because a method of effective implementation could not be found due to time constraints. Biofilm is however, still useful to model because the metabolite production reaches a maximum nearer the injection point since the bacterial transport is retarded by equilibrium adsorption. This mobilizes more of the oil and increases recovery. A chart diagramming the overall simulation process may be seen in Figure 5.3. The red ellipses signify all required user input; the reservoir geometry, fluid and rock properties, the initial state, and the simulation schedule. The reservoir geometry with the rock and fluid properties define the system model. The model may then be used with an initial state and a schedule to begin the simulation. Adjustments to the conservation equations are done dependent on the presence of the various MEOR options. We will now present the equations used for a simulation with biofilm. The microbe model equations are

$$\begin{aligned}
\mu_b &= \mu_{b,max} \frac{N}{K_b + N} \\
\mu_m &= \mu_{m,max} \frac{N}{K_m + N} \\
R_b &= \mu_b (B_f + B_a) Y_b \\
R_m &= \mu_m (B_f + B_a) Y_m \\
R_n &= -R_b - R_n \\
\mathcal{L} &= \frac{\omega_1 \omega_2 (B_f - B_a}{1 + \omega_2 (B_f - B_a)}.
\end{aligned}$$

Because the adsorption relationship is linear at these concentration values, this method of updating $\mathcal{L}$ is okay. The conservation equations are then

$$\begin{aligned}
\text{Water:} \quad & \frac{(S_w \phi \rho_w)^{n+1} - (S_w \phi \rho_w)^n}{\Delta t^n} + \textsc{div}(\rho_w \vec{u}_w)^{n+1} - q_w^{n+1} &&= 0 \\
\text{Oil:} \quad & \frac{(S_o \phi \rho_o)^{n+1} - (S_o \phi \rho_o)^n}{\Delta t^n} + \textsc{div}(\rho_o \vec{u}_o)^{n+1} &&= 0 \\
\text{Free Bacteria:} \quad & \frac{(S_w \phi \rho_w B_f)^{n+1} - (S_w \phi \rho_w B_f)^n}{\Delta t^n} + \textsc{div}(B_f \rho_w \vec{u}_w)^{n+1} - R_b - q_b^{n+1} &&= 0 \\
\text{Adsorbed Bacteria:} \quad & \frac{(S_w \phi \rho_w B_a)^{n+1} - (S_w \phi \rho_w B_a)^n}{\Delta t^n} - \mathcal{L} A_v S_w \phi &&= 0 \\
\text{Metabolites:} \quad & \frac{(S_w \phi \rho_w M)^{n+1} - (S_w \phi \rho_w M)^n}{\Delta t^n} + \textsc{div}(M \rho_w \vec{u}_w)^{n+1} - R_m &&= 0 \\
\text{Nutrients:} \quad & \frac{(S_w \phi \rho_w N)^{n+1} - (S_w \phi \rho_w N)^n}{\Delta t^n} + \textsc{div}(N \rho_w \vec{u}_w)^{n+1} - R_n - q_n^{n+1} &&= 0.
\end{aligned}$$

Adjustments to fluid properties are done prior to the formation of the conservation equations based on the type of metabolite which changes the Darcy velocity $\vec{u}_i$. The equations are then made into a linearized problem and a solution is found as usually done in MRST.

Chapter 6

Recreation of Results

With the MEOR model formulated and implemented for MRST, the first benchmark test to be made is to attempt to recreate results that have been achieved previously. This will determine if the model is behaving as expected and if the simulations can be useful. The first test will be attempting to recreate the results from Nielsen et al. (2010) [16] where biosurfactant is produced in a one-dimensional reservoir. Here the term one-dimensional refers to the direction of fluid flow, not the reservoir geometry. After this, the results from Lacerda et al. (2012) [7] will be reproduced. This is also a one-dimensional reservoir, however, in this case, the production of biopolymer is investigated.

6.1 Simulation of Biosurfactant

The work presented in Nielsen et al. (2010) [16], tests the effects of microbial produced biosurfactant on oil recovery. To simulate the effects on relative permeability, they used the capillary number method, Coats' interpolation method, and the Corey interpolation method. They also investigated the sensitivity of the parameters, varying the distribution coefficient $\mathcal{K}$, the maximum growth rate μ, and using different injection concentrations of microbes and nutrients. This work will not seek to recreate all these tests and results but will instead focus on using the Corey interpolation method with a single set of parameters. Additionally, in order to compare their simulations to each other more fairly, Nielsen et al. only alter S_{or} with their Corey interpolation technique. As this is the case, for comparison, this is the only relative permeability value that will be changed here. The total list of parameters may be found in Table 6.1. The type of surfactant being simulated is a very efficient one to demonstrate the potential of MEOR. With the initial IFT, σ, equal to 29 mN/m, the calculated IFT with this surfactant follows the equation,

$$\sigma^*(M_s) = 29\frac{-\tanh(1.5\times10^4 M_s - 0.2) + 1 + 10^{-4}}{-\tanh(-0.2) + 1 + 10^{-4}}.$$

The graph of this function dependent on surfactant concentration is drawn in blue in Figure 4.1. The reservoir is injected with water from one side at a rate of 800 m^3/day with nutrients at a concentration of 10^{-2} kg/m^3, and bacteria at a concentration of 0.5×10^{-2} kg/m^3. The bacteria has a maximum growth rate of 0.2/day. This low rate was chosen because anaerobic bacteria reproduce slower than aerobic bacteria, and because the reservoir is not an ideal environment for reproduction. The reservoir dimensions are 400 m × 100 m × 100 m, with a porosity of $\phi = 0.4$. That gives a total pore volume of 1,600,000 m^3. With the initial water saturation at 0.3, this means that the amount of oil in the reservoir at the start of recovery is 1,120,000 m^3. To measure the recovery factor, a ratio of the amount

Table 6.1: Parameters used for recreating the biosurfactant effects of Nielsen et al.

	Parameter	Value
Parameters from Nielsen	Y_b	0.82
	Y_m	0.18
	K_b	1 kg/m^3
	K_m	1 kg/m^3
	μ_b	0.2/day
	μ_m	0.2/day
	$\mathcal{K}$	1
	S_{or}	0.4
	S_{wi}	0.3
	kr_{owi}	0.8
	kr_{wor}	0.5
	n	2
	ρ_w	1000 kg/m^3
	ρ_o	800 kg/m^3
	η_w	1 cP
	η_o	3 cP
	σ	29 mN/m
	a	6
	l_1, l_2, l_3	$\{10^{-4}$, 0.2, $1.5 \times 10^4\}$
	Reservoir Dimensions	400 m × 100 m × 100 m
	$\Delta x \times \Delta y \times \Delta z$	1 m × 100 m × 100 m
	Volumetric Injection Velocity	800 m^3/day
	ϕ	0.4
	q_b	0.5×10^{-2} kg/m^3
	q_n	10^{-2} kg/m^3
Defined Parameters	$\mathbf{K}$	100 mD
	p	10^7 Pa
	Δt	1 day

of oil recovered to the original oil in place, (OOIP), will be used,

$$\%\ \text{OOIP} = \frac{\text{Oil recovered}}{\text{Original oil in place}} \cdot 100.$$

The simulations are run for a total of 2000 days with $\Delta t = 1$ day, this corresponds to injecting a complete pore volume into the reservoir. The original Δt used by Nielsen et al. was 1.2, we chose to alter that purely for simplicity. It is assumed that there are no bacteria present in the reservoir from before and that the viscosities of oil and water remain constant. The results are compared against those from a pure water flood and with the results from Nielsen et al.

6.1.1 Comparison and Analysis

With a pure water flood, our simulation predicts a recovery of 38% OOIP. This is slightly less than the result from Nielsen et al. which is 41% OOIP. The discrepancy may be explained by the fact that the model in Nielsen et al. uses a different method of solving the problem. They use a fractional flow function and the total velocity, this substitutes the phase velocities and removes both the pressure,

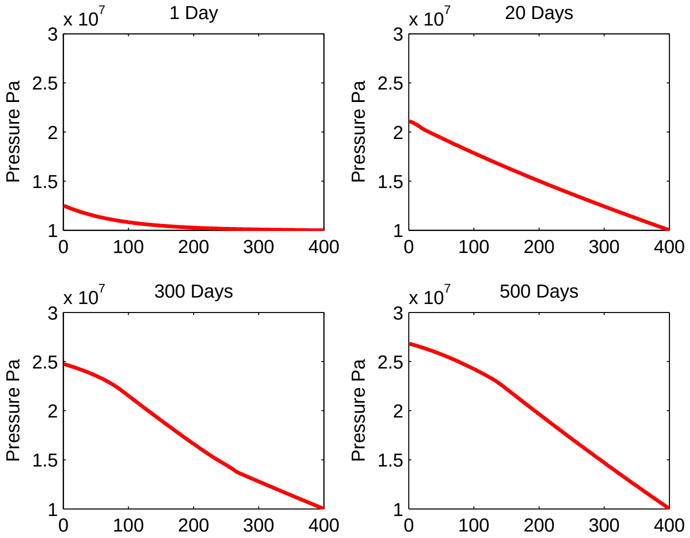


Figure 6.1: Pressure profile for the reservoir at different times. The pressure is initialized at 10^7 Pascal throughout the reservoir and a boundary condition enforcing this is present on the right side. This simulates a production well controlled by a bottom hole pressure. A constant volumetric injection rate of 800 m^3/day is used on the left.

p, and absolute permeability, $\mathbf{K}$, from the equations. As mentioned previously, MRST requires both of these values to be initialized before solutions may be found. The difference in these methods will therefore result in slightly different results. The pressure is initialized to 10^7 Pascals whereas the absolute permeability is set at a homogeneous 100 millidarcys throughout the reservoir. The absolute permeability will not change as the biosurfactant effects the relative permeability and there is no pore clogging. The pressure however, does change significantly throughout the simulation as can be seen in Figure 6.1. The reservoir wells are simulated in this setup as boundary conditions. On the left, there is a Neumann condition of a constant water flux of 800 m^3/day. The right side has a Dirichlet condition of a constant pressure of 10^7 Pascals. The Dirichlet condition is similar to production wells that have a minimum bottom hole pressure to be operative. In Figure 6.2, a comparison of the recovery profiles from the MEOR simulation and from solely a water flood may be seen. The MEOR simulation succeeds in recovering 71% OOIP, an increase of 33% OOIP or 85% more than the water flood. While this is certainly a significant improvement, it is still considerably less than the result from Nielsen et al. When simulating the surfactant effect with the Corey method, the incremental increase in recovery for their simulation is 38% OOIP. A quantitative comparison of the simulations from Nielsen et al. and the thesis model may be seen in Table 6.2. This table includes a column each for oil recovered

Table 6.2: Comparison of surfactant based MEOR simulations after one pore volume injection, 2000 days.

	Water Recovery %	MEOR Recovery %	Incremental %	Improvement %
Nielsen et al.	41	79	38	93
Thesis Model	38	71	33	85
Difference	-3	-8	-5	-8

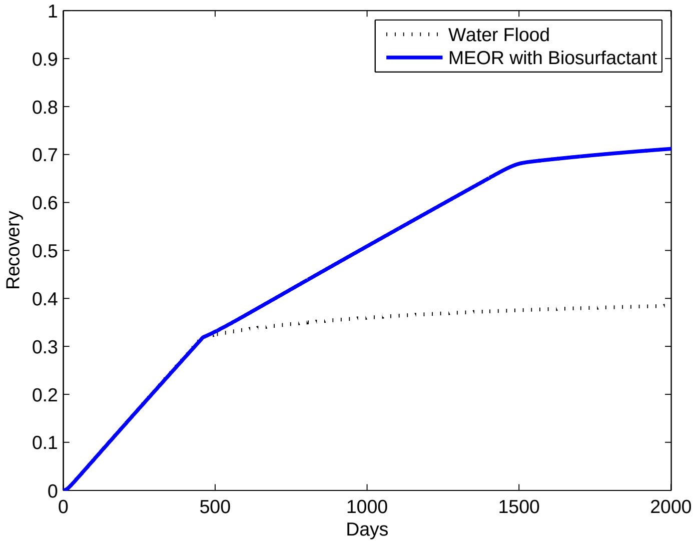


Figure 6.2: Oil recovery profile for both MEOR with surfactant and for a pure water flood for 2000 days. The MEOR flood succeeds in improving recovery over 85%.

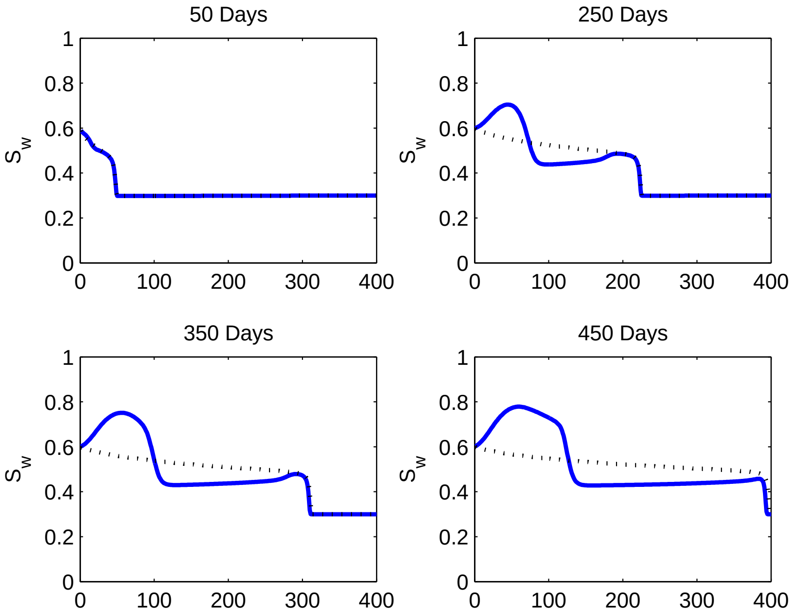


Figure 6.3: Saturation profile of MEOR recovery in the solid line at different times with recovery without MEOR shown in the dashed line. The biosurfactant forms an oil bank that increases oil recovery.

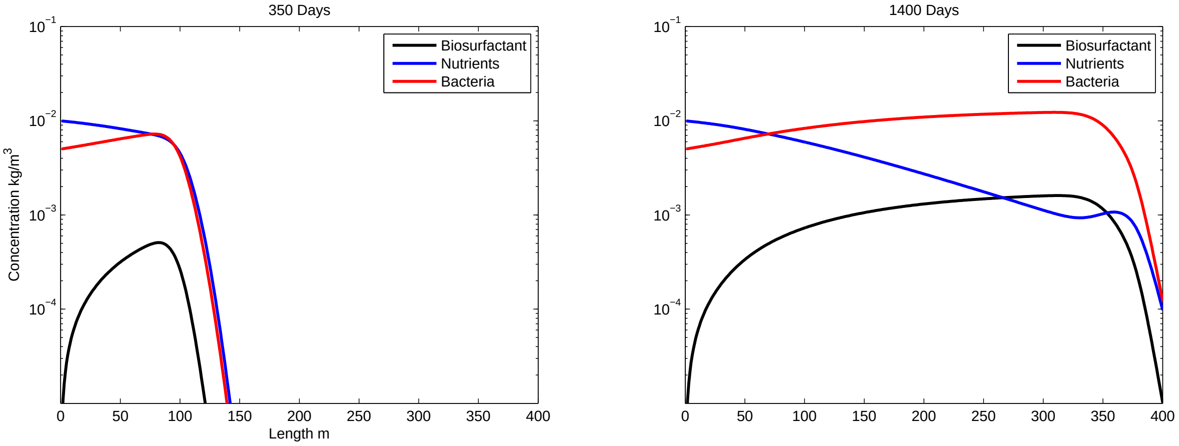


Figure 6.4: Concentration of bacteria, nutrients, and biosurfactant after 350 days and 1400 days of MEOR flooding.

from water flooding and from the MEOR simulation. It then presents columns for the incremental improvement and a relative improvement. That is,

$$\text{Incremental \%} = \text{Water Recovery \%} - \text{MEOR Recovery \%}$$

$$\text{Improvement \%} = \frac{\text{Incremental \%}}{\text{Water Recovery \%}}.$$

Results are presented for both the Nielsen et al. model and the thesis model. The differences between the two are also listed. Figure 6.3 shows the water saturation profile at 50, 250, 350, and 450 days. The MEOR flood is drawn in blue and the water flood in black. The figure clearly illustrates how the biosurfactant alters the recovery process. The creation of an oil bank and a second waterfront is what increases recovery. This is caused by the sharp decrease in IFT due to the high biosurfactant concentration. The total recovery initially follows the water flood and this is evident in Figure 6.2, with the recovery lines coinciding in the beginning. Then, after a little more than 450 days, the water recovery plateaus while the second front created by the biosurfactant continues recovery for the MEOR simulation, though not at as high a rate as before. After the second front reaches the other side of the reservoir, the recovery level for the MEOR simulation plateaus as well. How the concentrations of the bacteria, nutrient, and biosurfactant change over time may be seen in Figure 6.4. The concentrations are shown at 350 and 1400 days or 0.175 and 0.7 pore volume injections. The bacteria displays steady growth that begins to slow down when the nutrient concentration becomes depleted. The nutrient concentration is not monotonically decreasing however. There exists a portion of the curve where the concentration begins to rise again and this can be seen clearly after 1400 days. The figure shows that the minimum concentration of nutrients coincides with the maximum bacteria and metabolite concentrations. Logically, more microbes will consume nutrients at a faster rate. As the growth and production rates are variable based on the Monod model equation (4.1), when the nutrient concentration is nearly depleted, the growth rates will drastically decrease. Therefore it follows that the presence of this abnormality in the curve is the result of a decrease in the growth rate due to a lack of nutrients. The biosurfactant concentration quickly increases and then begins to slow down when the nutrients are used. This is a trait of the Monod model. The growth of bacteria and production of metabolites is only constrained by the presence of the limiting nutrient. It can be seen that the injection concentrations of nutrients and bacteria is at the proper level for the nutrient to penetrate through the entire reservoir.

When considering the differences between the solution methods used by Nielsen et al. and this thesis, the discrepancies in the results are understandable and explainable. The method of simulating the reservoir used by MRST includes pressure and compressibility which can hinder the flow of the phases in the reservoir. This is why the thesis model results in a lower recovery level after water and MEOR flooding, and a lower incremental recovery. There is still a significant improvement due to the surfactant and that is what is most important. Using all the parameters from Nielsen et al. and a few others that are required by MRST, we have developed a model to simulate surfactant-based MEOR successfully. What remains to be seen is if the model for polymer-based MEOR is equally triumphant.

6.2 Simulation of Biopolymer

In the article by Lacerda et al. (2012) [7], the use of a polymer producing bacteria in oil recovery is investigated. The biopolymer serves to increase the sweep of the water flood and force more oil out of the reservoir. The use of different equations to simulate the biopolymer effects was tested as well as different modifications of bacterial growth modeling. We will use both equation (4.8), the power law, and equation (4.7), the parabolic law, to simulate the viscosity changes and then compare results. There will be no adsorption effects simulated as these are not present in Lacerda et al. Many of the parameters presented in Lacerda et al. are not strictly specified in order to present the results in as general a manner as possible. For instance, the reservoir dimensions are not listed but are instead scaled variables with the length of the reservoir equal to 1. While that does not pose much of an issue, the fact that the injection concentrations of both nutrients and bacteria are not listed does. Without knowing the ratio or quantity used by Lacerda et al., the results will be necessarily different. Nevertheless, an attempt will be made and several of the parameters will be identical to those used in the previous section. What is defined from Lacerda et al. is the initial fluid viscosities and relative permeability parameters, but not their densities. For the microbe model, the half saturation constants and yield values are present. The maximum growth rate however, is also difficult to determine. It is given as 8.237 (vpi^{-1}). The unit of vpi is never explained. If this is meant to indicate a maximum growth rate of 8.237 per pore volume injection, that would correspond to a growth rate of 0.004/day in our reservoir setting. As that value is incredibly low, the injection rate of bacteria would need to be unrealistically high to produce sufficient amounts of biopolymer. It does seem that the injection ratio of nutrients to bacteria is 5:1. This is determined from Figure 3 in Lacerda et al. which presents graphs of the bacteria and biopolymer concentrations. It seems these values are scaled and would be presented in our setting as $\frac{B}{q_b}$ and $\frac{M}{q_b}$. Using that ratio in our simulations achieves maximum values of $\frac{B}{q_b} = 3.5$ and $\frac{M}{q_b} = 2.5$, which appears to also be true for Lacerda et al. With that ratio and taking into account how much biopolymer is needed to produce an effect on the flow, we choose a maximum growth of 0.01/day with injection concentrations of 1 kg/m^3 and 0.2 kg/m^3 for nutrients and bacteria respectively. These values for modeling bacteria are significantly different from those used in Nielsen et al. and it is unknown how realistic they are. With this injection rate, they indicate that an extra 960 kg of material would need to be injected into the reservoir each day. All parameters used may be seen in Table 6.3. Again, the solutions in Lacerda et al. are found using fractional flow and therefore the absolute permeability and pressure are not given. There is no metabolite partitioning or biofilm formation.

6.2.1 Comparison and Analysis

As the quantitative results of the biopolymer simulation are found with dubious parameters, a comparison with Lacerda et al. is done simply to provide a reference point. When using the power law, the

Table 6.3: Parameters used for recreating the biopolymer effects of Lacerda et al.

	Parameter	Value
Parameters from Lacerda	Y_b	0.5
	Y_m	0.5
	K_b	0.5 kg/m^3
	K_m	0.5 kg/m^3
	η_w	0.5 cP
	η_o	5 cP
	S_{wi}	0.16
	S_{or}	0.23
	kr_{wor}	0.3
	kr_{owi}	0.7
	n	2
Defined Parameters	Reservoir Dimensions	400 m × 100 m × 100 m
	$\Delta x \times \Delta y \times \Delta z$	1 m × 100 m × 100 m
	Δt	1 day
	Volumetric Injection Velocity	800 m^3/day
	q_b	0.2 kg/m^3
	q_n	1 kg/m^3
	μ_b	0.01/day
	μ_m	0.01/day
	ϕ	0.4
	$\mathbf{K}$	100 mD
	p	10^7 Pa
	ρ_w	1000 kg/m^3
	ρ_o	800 kg/m^3

Table 6.4: Comparison of polymer based MEOR simulations using the power law after one pore volume injection, 2000 days. Results are shown up to two decimal places.

	Water Recovery %	MEOR Recovery %	Incremental %	Improvement %
Lacerda et al.	54.33	64.64	10.31	18.98
Thesis Model	54.35	62.34	7.99	14.71
Difference	0.02	-2.30	-2.32	-4.27

Table 6.5: Comparison of polymer based MEOR simulations using the parabolic law after one pore volume injection, 2000 days. Results are shown up to two decimal places.

	Water Recovery %	MEOR Recovery %	Incremental %	Improvement %
Lacerda et al.	54.33	69.74	15.41	28.36
Thesis Model	54.35	64.56	10.21	18.79
Difference	0.02	-5.18	-5.20	-9.57

recovery is only increased by 7.99% over 2000 days of water flooding. With the parabolic law, recovery is improved by 10.21%. This may be seen in Figure 6.5. Curiously, both the model from Lacerda and the thesis model have nearly the exact same recovery value from only water recovery, differing only 0.02%. The results in Lacerda et al. were presented with 2 decimal places so we will use the same convention in this section for comparison. The MEOR recovery results from the thesis model are each less than their counterparts from Lacerda et al. However, when using the power law to simulate the biopolymer effects, Lacerda et al. report recovery of 64.64% OOIP as opposed to our result of 62.34% OOIP. Similarly, the difference in incremental recovery when using the parabolic law is 5.18%. This means that the results for this simulation are actually closer than they are for the biosurfactant. The reason for the discrepancies are more understandable in this situation however, as many of the parameters were not given in addition to the difference in obtaining solutions. Comparisons of the results can be found in Tables 6.4 and 6.5. This presents the results in the same manner as the previous section. Also in Figure 6.5, the pressure changes due to the MEOR flood may be seen. The only thing different between the two MEOR models is the method of changing the viscosity, though it is clear that this affects other aspects as well. The most important difference is seen in the saturation profiles in Figure 6.6. It can be observed that the typical second water front often seen in polymer EOR simulations is not present when using the power law. Instead there is only a wave. The second front does form when using the parabolic law however and the first front is not slowed down as much when compared to the power law flood. The effect of this may be seen in the recovery profile where the power law flood is the most effective after the first front reaches the production well but then is overtaken by the parabolic law recovery. The reason for this is that the power law is highly effective with low concentrations of biopolymer though it reaches a maximum value after time. The calculated viscosities at different times may be seen in Figure 6.7. It can be seen that the power law changes the water viscosity quickly and at the same time creates a more gradual transition between areas with and without biopolymer. Conversely, the parabolic law takes longer to demonstrate a significant change in viscosity but reaches a much higher level with a relatively steep slope. This is why the second water front is formed.

In this simulation, the nutrients are all consumed about halfway through the reservoir. At that point, the bacteria stops multiplying and producing any more biopolymer. Thus, the maximum effect is reached. This may be seen in Figure 6.8. While there are some differences between where this happens in the reservoir depending on which viscosity equation is used, the general behavior is the same. The bacteria model also displays the same traits as the recreation of Nielsen et al. except in

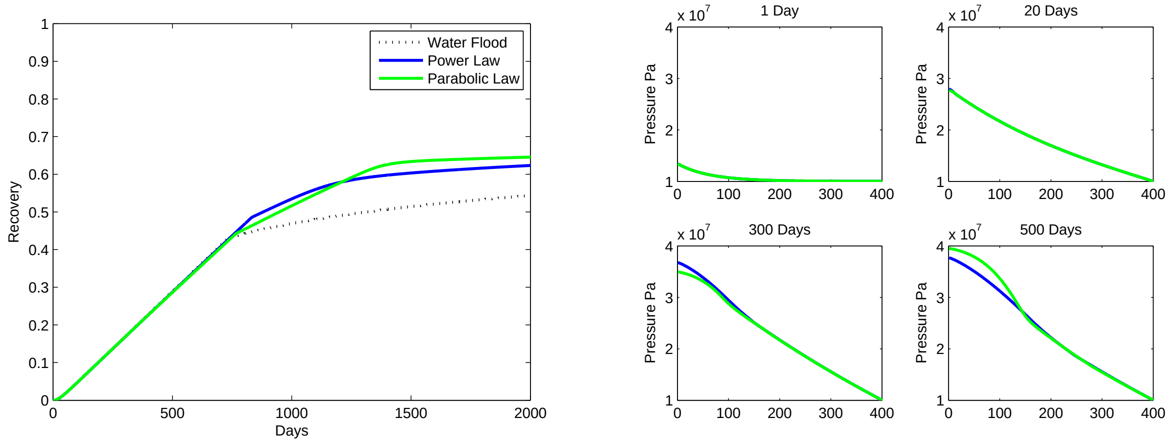


Figure 6.5: Oil recovery profile and pressure graphs for both the power law, blue, and parabolic law, green. The MEOR flood succeeds in improving recovery about 15%. It is clear that the different methods of changing viscosity have an effect on the pressure as well.

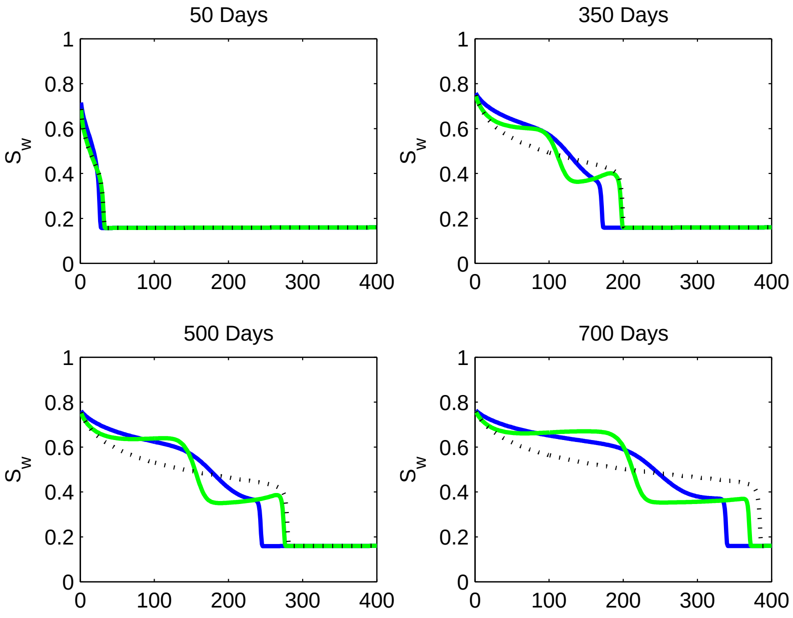


Figure 6.6: Saturation profile of recovery at different times with water flood recovery shown in the dashed black line. MEOR recovery using the power law is in blue and with the parabolic law in green.

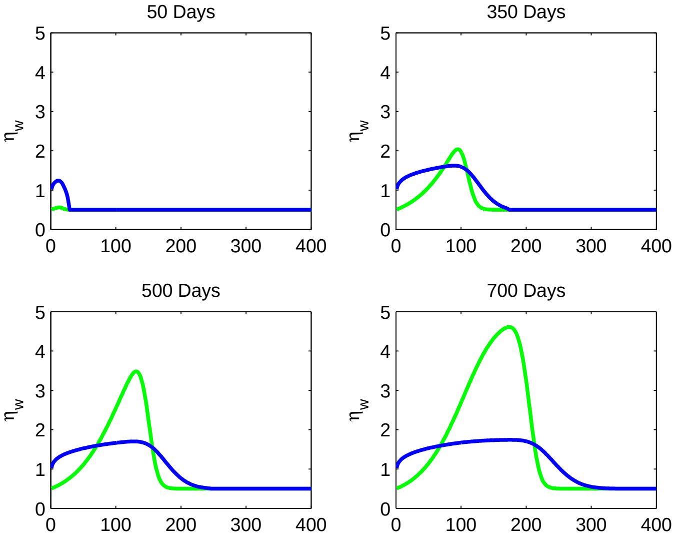


Figure 6.7: Calculated water-polymer mixture viscosity values for the power law in blue and the parabolic law in green. Initial water viscosity is 0.5 cP and oil viscosity is 5 cP.

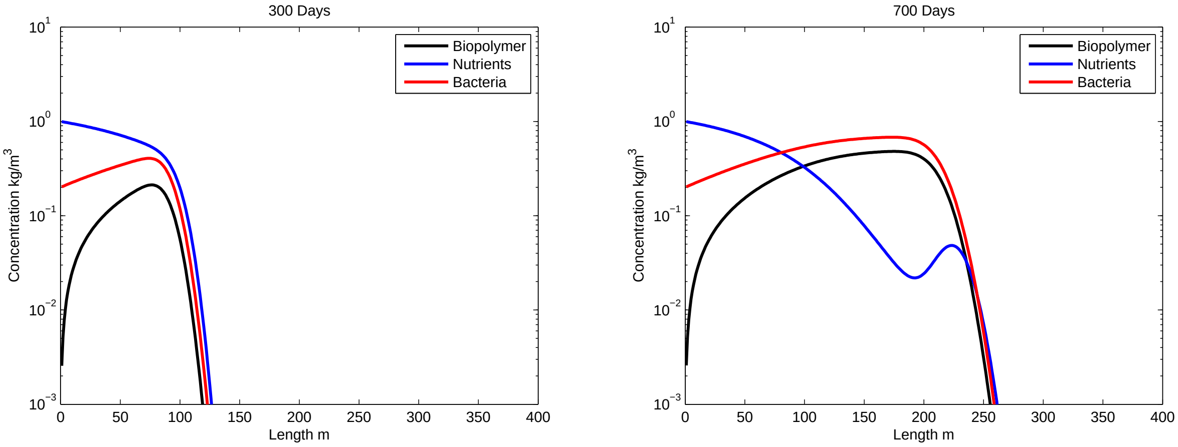


Figure 6.8: Concentration profiles of bacteria, nutrients, and biopolymer after 350 and 900 days of MEOR flooding with the power law. There are differences depending on which viscosity equation is used though the general behavior is the same.

Table 6.6: Parameters used for biofilm simulations

Parameter	Value
ω_1	0.001
ω_2	0.0017
A_v	3×10^5
ρ_b	1000

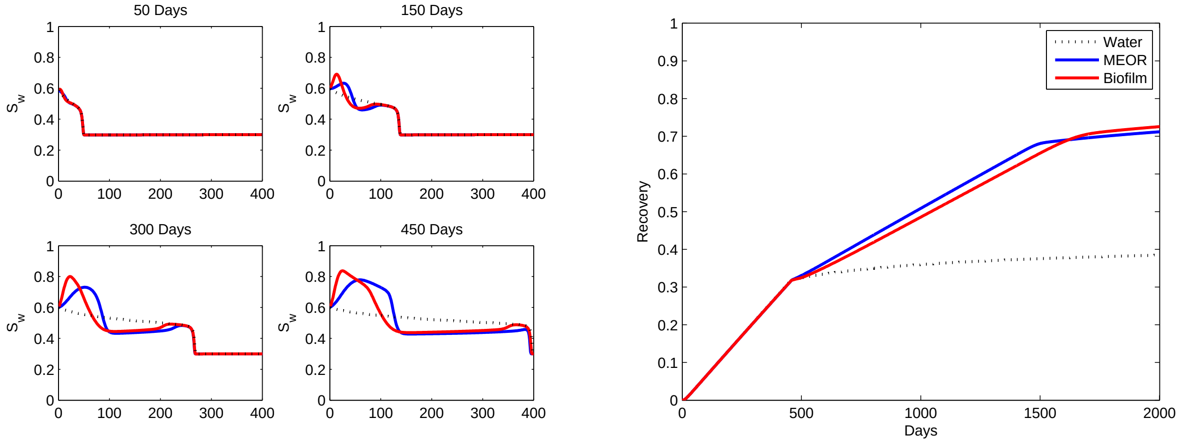


Figure 6.9: Saturation and recovery results for the Nielsen et al. simulation with biofilm. The biofilm simulation is in red, while the recovery without biofilm is in blue. The black dotted line is recovery with only water.

this instance, equal amounts of bacteria and metabolite are produced as the yield values are equal. The polymer-based MEOR simulation does improve recovery as expected. Though, it is unknown how realistic some of the parameters used are.

6.3 Biofilm Results

We would now like to investigate the results from the previous two sections if biofilm was allowed to form and restrict the transport of bacteria in the reservoir. This will serve to test if the biofilm model is performing correctly and ensure that it does in fact improve recovery. Both simulations are ran identically to the previous sections with the exception of the biofilm necessary parameters seen in Table 6.6. The Langmuir parameter ω_1 is defined to allow a maximum biofilm formation of 75% of the pore volume while ω_2 is defined to ensure that approximately half the bacteria will ultimately adsorb.

6.3.1 Biosurfactant with Biofilm

We first look at the simulation ran for recreating Nielsen et al. with biofilm and find that recovery is increased by 1.36% OOIP reaching a total of 72.55% OOIP recovered. The effect of the biofilm on the saturation profile and recovery is seen in Figure 6.9. The recovery without biofilm initially produces more oil but plateaus sooner and biofilm recovery surpasses it. The reason for this can be seen in Figure 6.10. This figure displays both the biosurfactant concentration, M_s, as well as the corresponding residual oil saturation value, S_{or}, for both models with and without biofilm. Recall

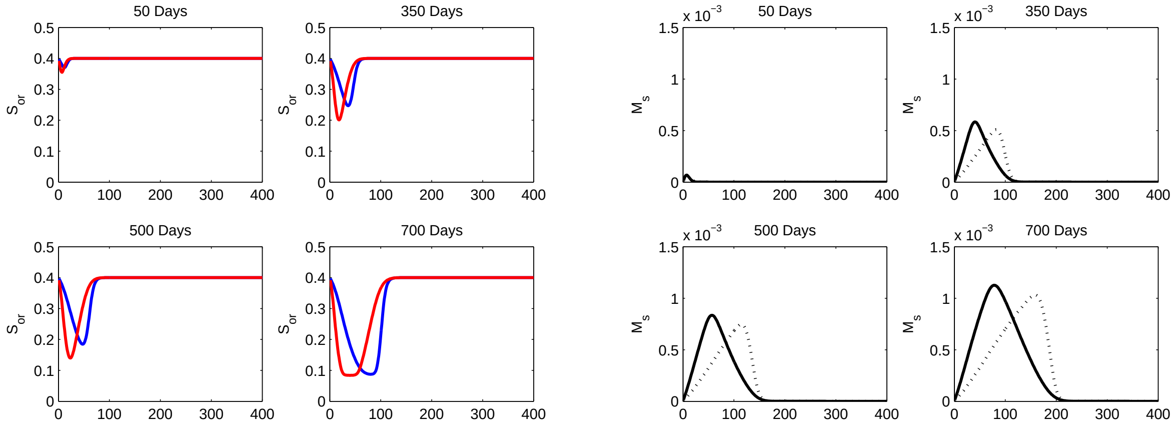


Figure 6.10: The residual oil value S_{or}, used for calculating relative permeability visualized in the graphs on the left and the corresponding biosurfactant concentrations M_s, on the right. The blue line on the left and the dashed black line on the right represents the simulation without biofilm.

that the residual oil saturation is the parameter changed by the Corey interpolation technique we are using to simulate the effect of biosurfactant. The model with biofilm reduces this value faster and nearer the injection point which leads to more oil being recovered overall. It can be seen that after 700 days, the model without biofilm has reduced S_{or} to nearly the same level further ahead in the reservoir which leads to more recovery initially. The Langmuir adsorption slows the transport of bacteria through the reservoir which consequently slows the metabolites. The nutrients however are not affected and this actually leads to a large portion being transported with no bacteria to consume them. This is displayed in the graphs of Figure 6.11. A significant amount of nutrients is traveling with no use to the bacteria. In a reservoir with indigenous bacteria, these would in all likelihood be consumed before reaching the production well.

6.3.2 Biopolymer with Biofilm

We now compare the results from the recreation of Lacerda et al. to a simulation ran with biofilm. The parabolic equation for viscosity change, equation (4.7), is used. The recovery is increased in this case by 1.51% OOIP to a total of 66.07% OOIP. Similar to the last simulation, the model without biofilm is initially more effective but then is overtaken after recovery plateaus. This is seen in Figure 6.12. Also, the well defined second water front is missing from the biofilm simulation because of how the biopolymer is produced. Whereas the simulation without biofilm gradually increases the metabolite concentration with the maximum near the front of the water flood, the biofilm simulation stimulates metabolite production at the beginning of the reservoir. Since the bacteria are transported at a reduced rate because of adsorption, the metabolite concentration decreases much more gradually in the front. How this then effects the viscosity change is seen in Figure 6.13. Again the effect is seen faster and nearer to the injection point which eventually increases overall recovery. The behavior of the rest of the substances in the reservoir is seen in Figure 6.14. Because the metabolite yield Y_m is 0.5 in this model, there is a much higher concentration when compared to the Nielsen simulation where $Y_m = 0.18$. There is again a substantial amount of nutrients that are not used.

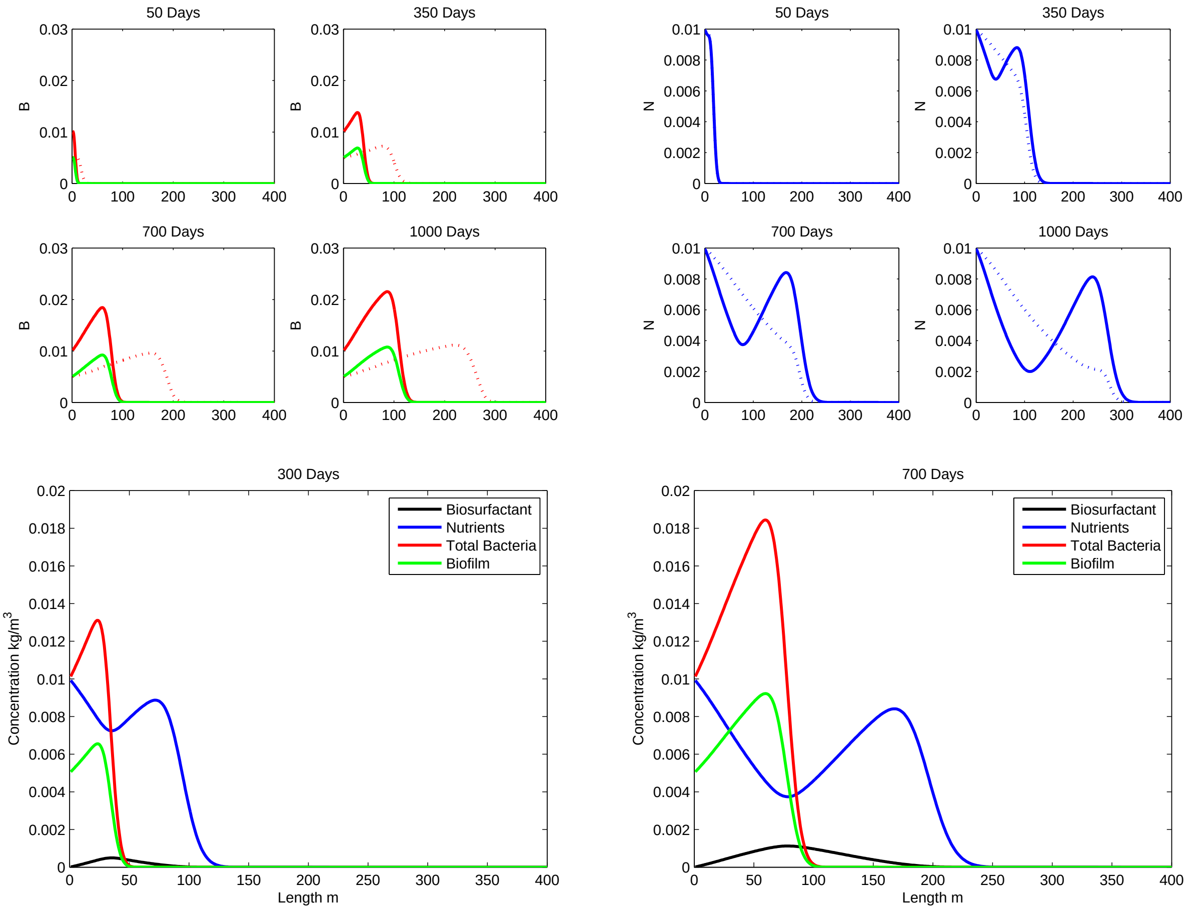


Figure 6.11: In the top left, a graph of bacteria concentration with red representing the total concentration with biofilm and the green representing the biofilm. The dashed red line is from the simulation without biofilm. Similarly, on the top right is the nutrient concentration, with the dashed line representing no biofilm. The figures on the bottom show how substances change in the biofilm model.

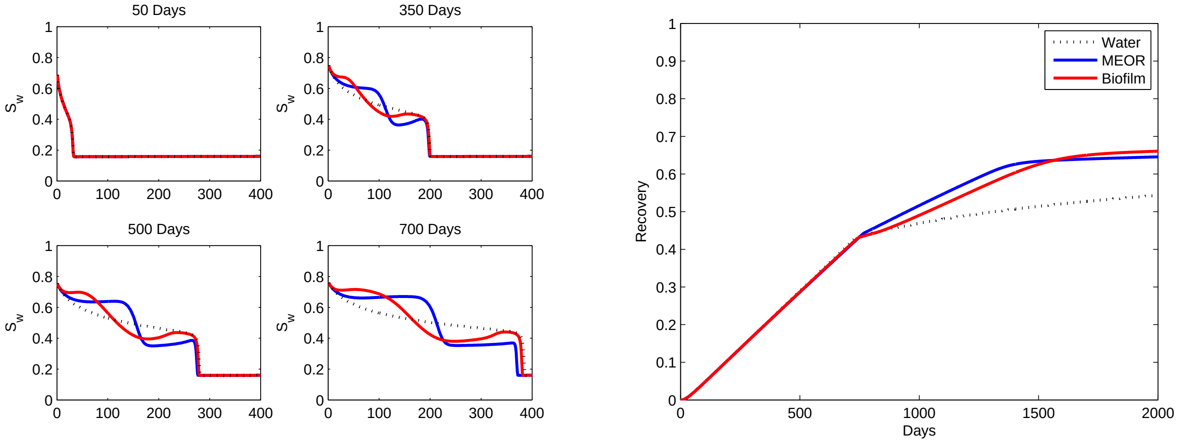


Figure 6.12: Saturation and recovery results for the Lacerda et al. simulation with biofilm. The biofilm simulation is in red, while the recovery without biofilm is in blue. The black dotted line is recovery with only water.

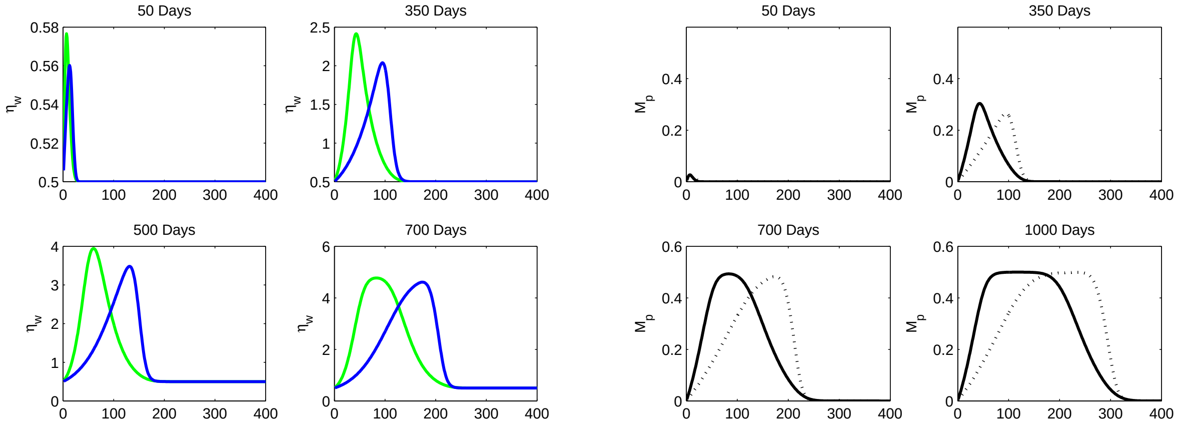


Figure 6.13: The new water viscosity value η_w is shown on the right. The biofilm model in green and without biofilm in blue. The corresponding biopolymer concentrations are on the right with the dashed black line representing the simulation without biofilm.

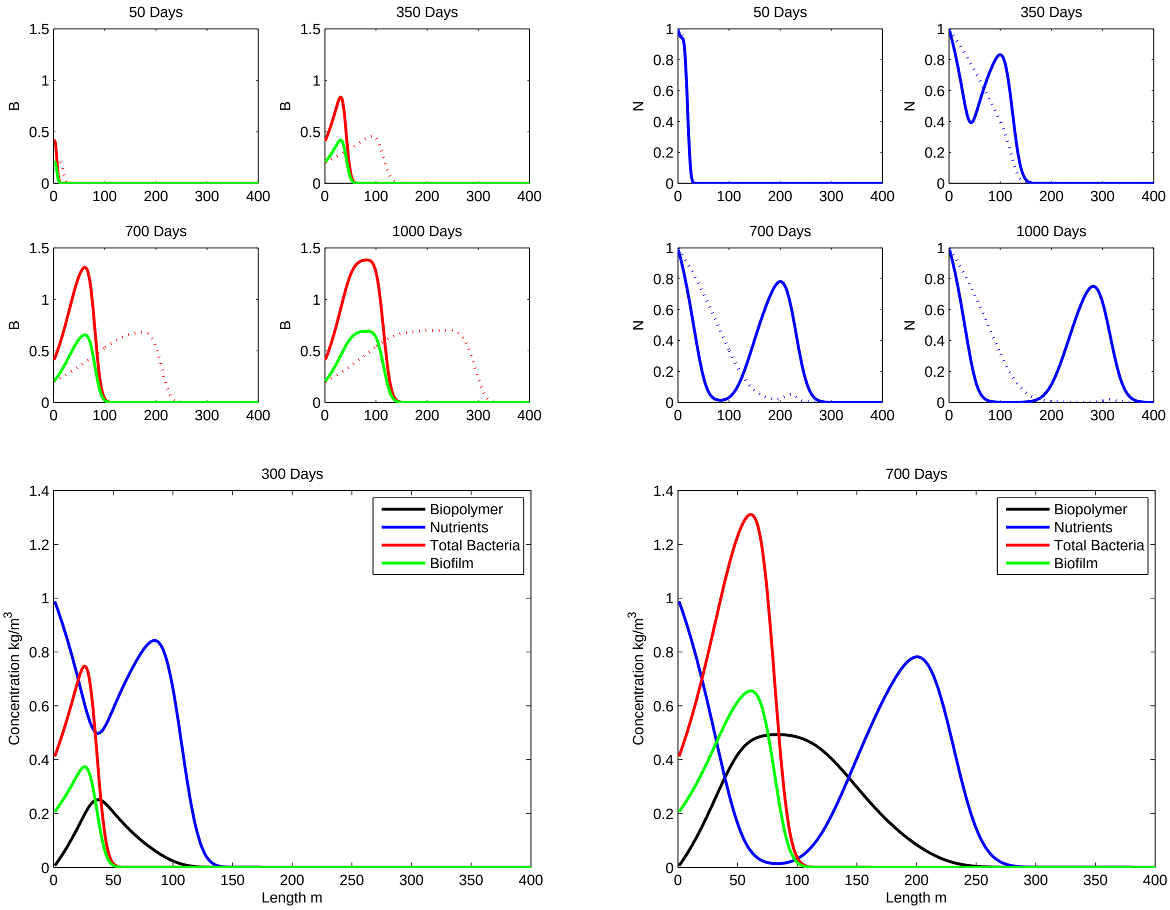


Figure 6.14: In the top left, a graph of bacteria concentration with red representing the total concentration with biofilm and the green representing the biofilm. The dashed red line is from the simulation without biofilm. Similarly, on the top right is the nutrient concentration, with the dashed line representing no biofilm. The figures on the bottom show how substances change in the biofilm model.

6.4 Evaluation

These simulations have shown that our model does display the qualitative effects we desire in much the same way as other similar studies have produced before. The bacteria consume nutrients, multiply, and produce a metabolite to improve the effect of the water flood. The recovery of oil is then increased a significant amount. The quantitative results are different, though this is the result of alternate solution methods and the use of different parameters. The addition of bacterial adsorption to the simulation produces metabolites nearer the injection boundary and improves recovery. This is however only a highly simplified test of one-dimensional flow through a homogeneous, box-shaped reservoir. While useful for demonstrating the concept of MEOR, it ignores the complexities of reservoirs which would actually increase the value of EOR techniques.

Chapter 7

Thief Zone Simulation

Thief zones have been mentioned previously as one of the main reasons that secondary recovery produces only a fraction of the oil present. Strategies to negate their effect include the drilling of additional injection wells in less permeable areas, and also polymer flooding. Therefore, we will investigate how MEOR with biopolymer producing bacteria can also be effective. In addition to the biopolymer reducing the mobility of the water, the biofilm formation would also slightly decrease the permeability of the thief zone. The simulation reservoir is again box-shaped but now two-dimensional flow is allowed. This follows logically as thief zones are not able to be simulated in a single dimension. There is a channel in the center of the reservoir which is three times as permeable as the outer region and the porosity is increased from 0.3 to 0.35. The reservoir may be seen in Figure 7.1 with the thief zone displayed in red. This simulation will once again be purely to demonstrate how the biopolymer aids recovery conceptually. Biosurfactant producing bacteria will also be studied to investigate how thief zones impact their use. The wells are simulated through the use of same boundary conditions previously used. The full list of parameters is found in Table 7.1. The viscosity will be changed by the parabolic equation (4.7), and the same injection concentrations used in the previous biopolymer test will be used.

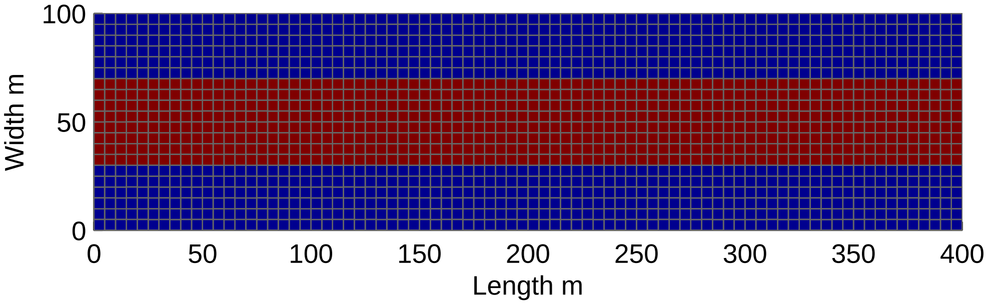


Figure 7.1: Thief zone simulation reservoir. The red channel in the center has a porosity of 0.35 and a permeability of 300 millidarcys. The blue area has porosity 0.3 and permeability 100 millidarcys.

Table 7.1: Parameters used for thief zone testing.

Parameter	Value
Reservoir Dimensions	400 m × 100 m × 100 m
$\Delta x \times \Delta y \times \Delta z$	5 m × 5 m × 100 m
Volumetric Injection Velocity	800 m^3/day
Δt	2.5 days
p	10^7 Pa
$\mathbf{K}$	100 mD and 300mD in the thief zone
ϕ	0.3 and 0.35 in the thief zone
ρ_w	1000 kg/m^3
ρ_o	800 kg/m^3
η_w	0.5 cP
η_o	7 cP
β_w	1
β_o	1
S_{or}	0.23
S_{wi}	0.16
kr_{owi}	0.7
kr_{wor}	0.3
n	2
$\mu_{b,max}$	0.2/day
$\mu_{m,max}$	0.2/day
Y_b	0.5
Y_m	0.5
K_b	1 kg/m^3
K_m	1 kg/m^3
$\mathcal{K}$	1
σ	29 mN/m
a	6
l_1, l_2, l_3	$\{41\times10^{-4}$, 2, 180$\}$
q_b	0.5×10^{-2} kg/m^3
q_n	10^{-2} kg/m^3

7.1 Recovery and Saturation

The change in porosity from the other tests means there is now less total pore volume. The OOIP is now 1,075,200 m^3. With an injection rate of 800 m^3/day, one PVI corresponds to 1,344 days of injection. Because of the thief zone, the initial breakthrough of the water front without biopolymer occurs much sooner than this, after approximately 400 days. After that happens, the water flood continues through the other areas, producing at a slower rate comparatively. This is also true for the biosurfactant recovery. The biopolymer reduces the mobility of the water flood and therefore the breakthrough of the front occurs much later. This may all be interpreted from Figure 7.2. After 1 PVI, recovery with only water attains 42.21% OOIP, recovery with biosurfactant 48.36% OOIP, and 55.54% OOIP with biopolymer. Stopping here however is a disservice to the biosurfactant. The simulations are ran for a total of 3,000 days or nearly 2.25 PVI. This allows the biosurfactant recovery to surpass the biopolymer by 0.5% OOIP. After 3000 days of injection, water based recovery produces just under half of OOIP while both MEOR recoveries increase that result by ten to 59% OOIP. This

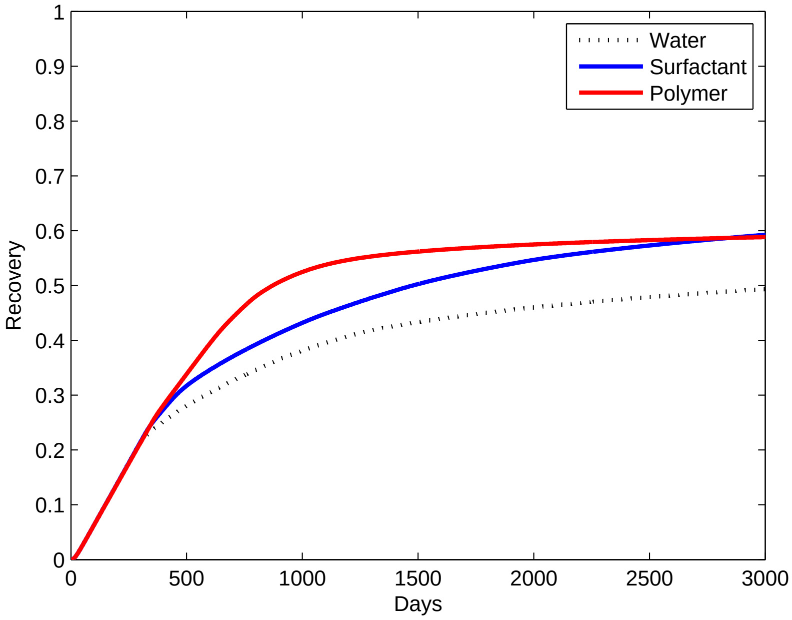
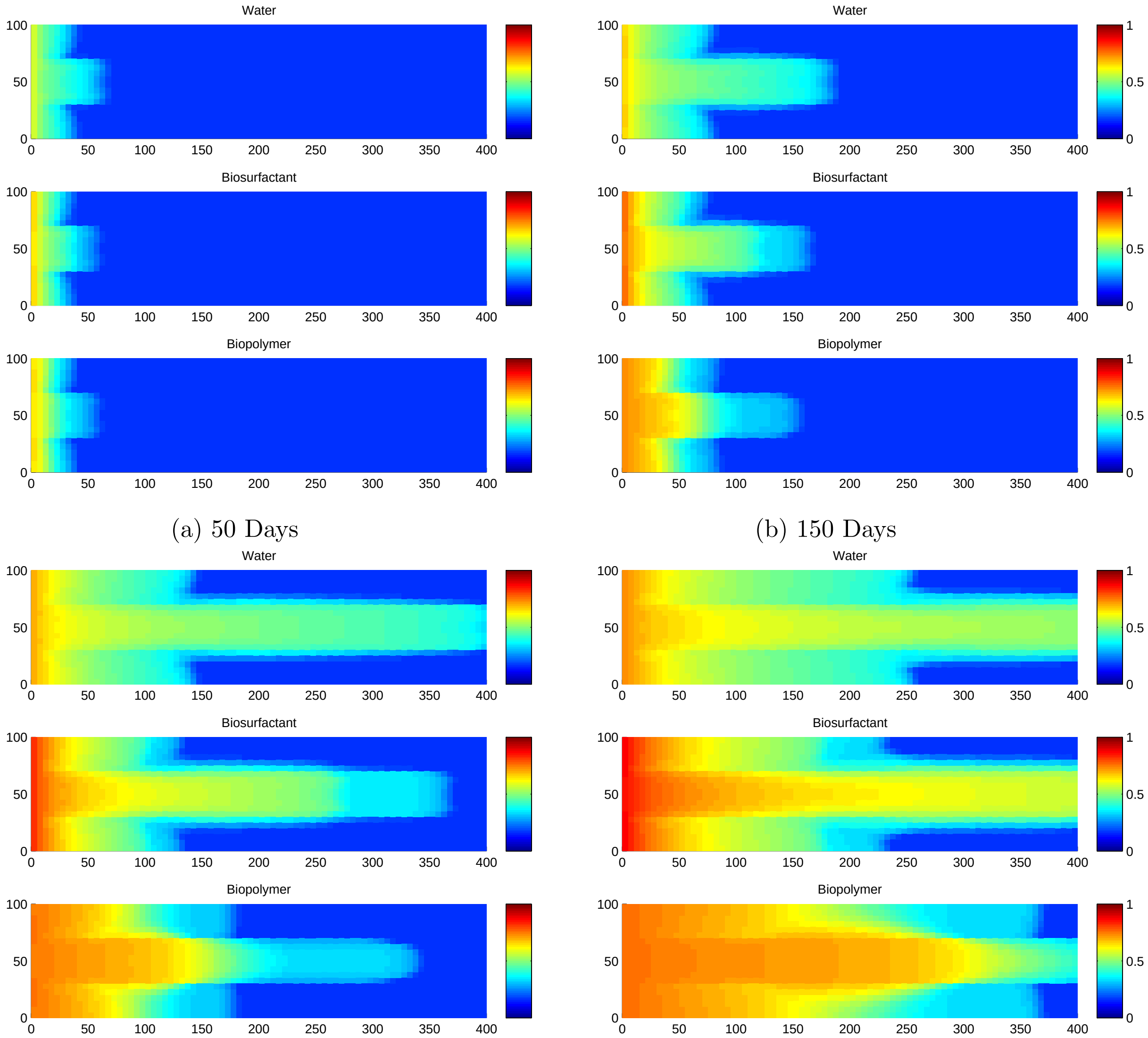


Figure 7.2: Recovery of OOIP in thief zone test with only a water flood, MEOR with biosurfactant, and MEOR with biopolymer.

Table 7.2: Recovery percentages of OOIP after several different days of thief zone simulations

	500	1000	1500	2000	2500	3000
Water	28	38	43	46	48	49
Biosurfactant	32	43	50	55	57	59
Biopolymer	34	52	56	57	58	59

is an improvement of 20%. More significantly is the speed with which this is accomplished with the use of biopolymer. After just 1000 days, recovery with biopolymer is already 52% OOIP. Results for each simulation at different times may be seen in Table 7.2. The saturation profiles of the simulations display quite interesting differences and we have therefore chosen to present several images from these. The first, Figure 7.3, shows the early stages of the simulation, from 50 to 750 days. After 50 days, there is virtually no difference between the three simulations and it is clear that the thief zone is creating a preferential path for fluid flow. The first discrepancies appear after 150 days. The biosurfactant simulation develops a very high saturation level directly at the injection boundary by decreasing the residual oil value. Since S_{or} remains unchanged in the biopolymer simulation, the saturation level does not become as high. The water saturation in the biopolymer simulation is higher than the level in the water flood and is more prevalent than the high concentration area of the biosurfactant simulation. After 350 days, the water flood reaches the production boundary through the thief zone. Outside of the thief zone, the biopolymer simulation is advancing the water front quicker than the other two simulations. This is clear after 750 days when the water front is past 350 meters into the reservoir for the biopolymer simulation and is at about 250 meters for the other simulations. The biopolymer also creates a more defined path through the thief zone. The flood in the other two simulations disperses outside of the thief zone boundary slightly. This does occur in the biopolymer simulation



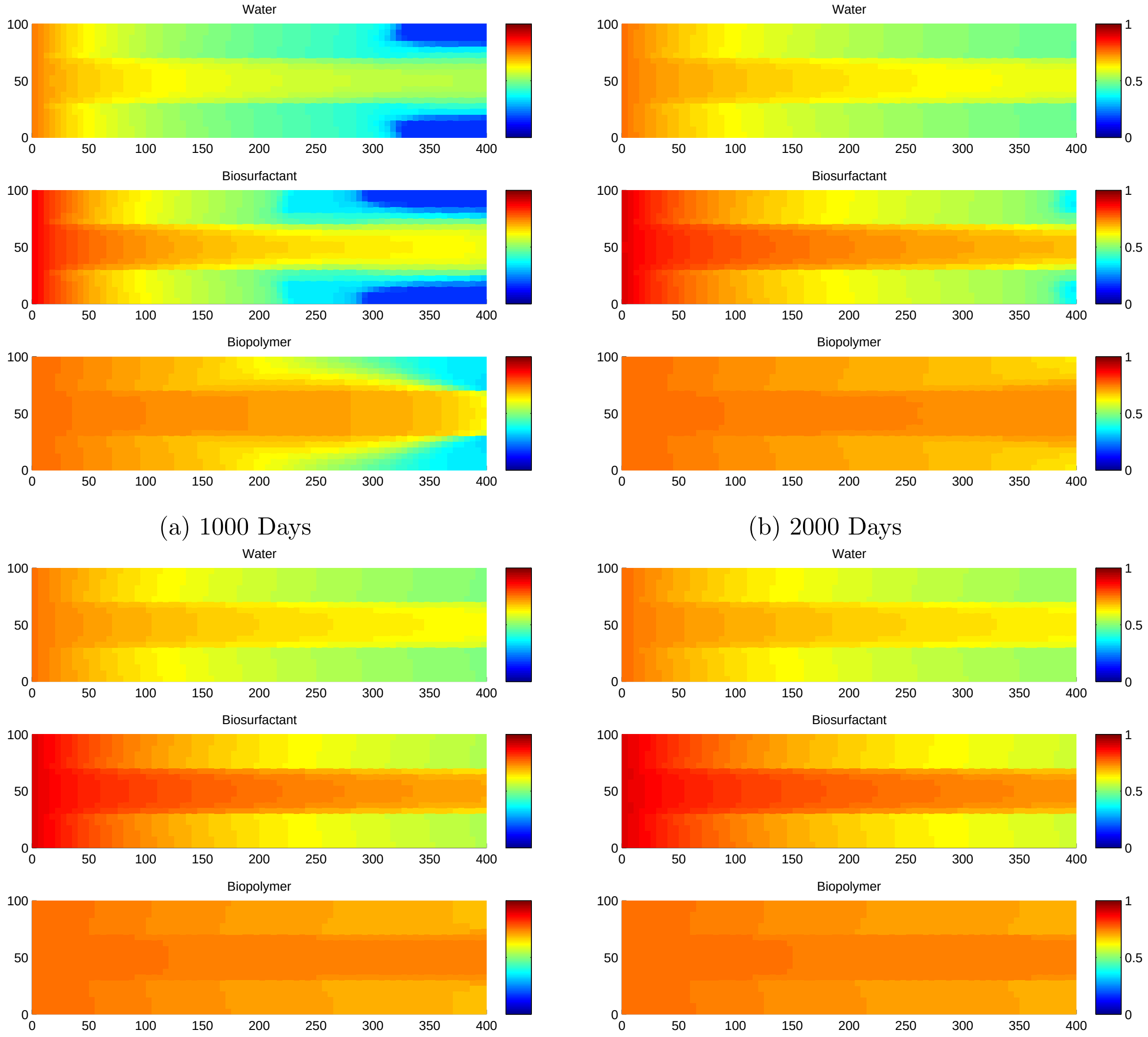
(a) 50 Days

(b) 150 Days

(c) 350 Days

(d) 750 Days

Figure 7.3: Saturation graphs from the different simulations in the early stages of injection.



(a) 1000 Days

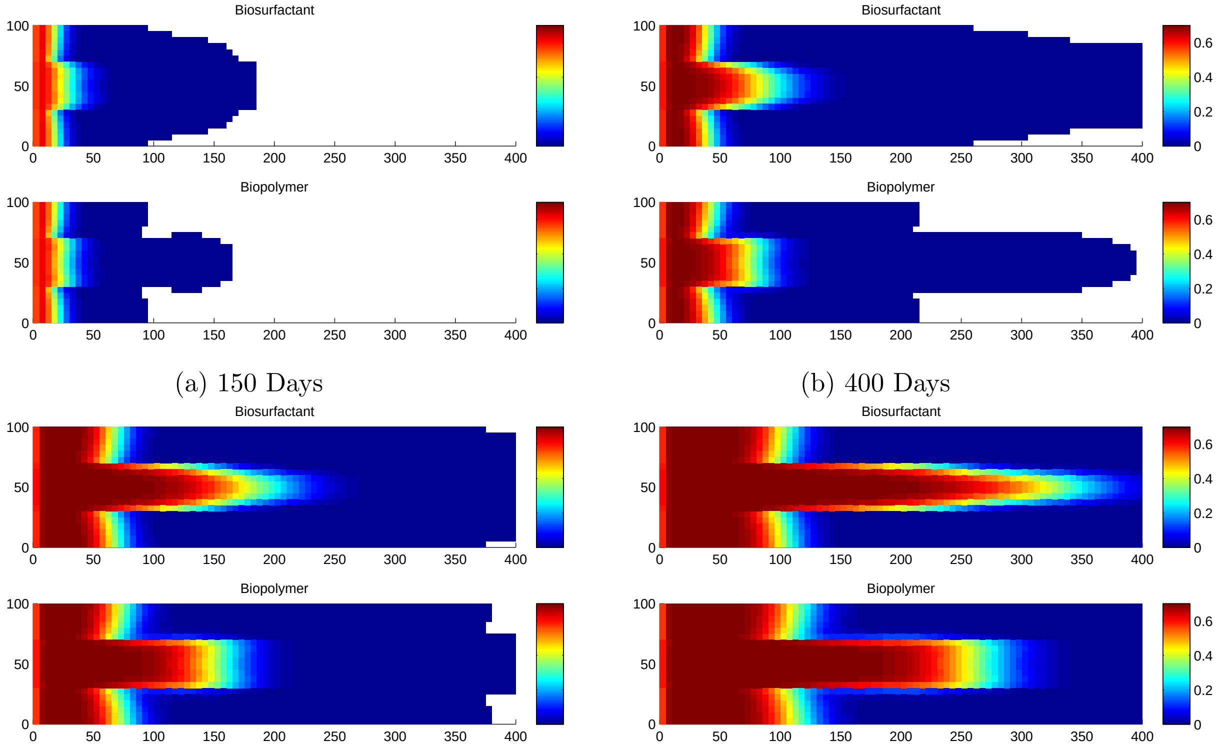
(b) 2000 Days

(c) 2500 Days

(d) 3000 Days

Figure 7.4: Saturation graphs from the different simulations in the later stages of injection.

as well but much further back in the reservoir. This is because the mobility of the water is reduced by the biopolymer which decreases the speed with which it travels through the thief zone. This is not the case in the other two simulations so the water saturation level in the thief zone is higher. So much so that it is easier for the water to flow outside the thief zone than remain in it. As for the saturation levels, the biopolymer has a much more uniform distribution, increasing volumetric sweep of the reservoir as advertised. The simulation with only water highly favors the thief zone. This is also true of the biosurfactant simulation. The distribution pattern is quite similar though the saturation values are higher. The differences after 1000 days seen in Figure 7.4 visualize how the techniques affect water flow inside the reservoir. Both water fronts, inside and outside the thief zone, have reached the outflow boundary for the biopolymer flood after 1000 days. At 2000, 2500, and 3000 days, there is not much difference in the biopolymer graphs. This matches the quantitative results as only a 1% increase in recovery happens between each figure. The biosurfactant figures show a brighter red color in the thief zone and near the injection boundary that corresponds to the reduced



(a) 150 Days

(b) 400 Days

(c) 750 Days

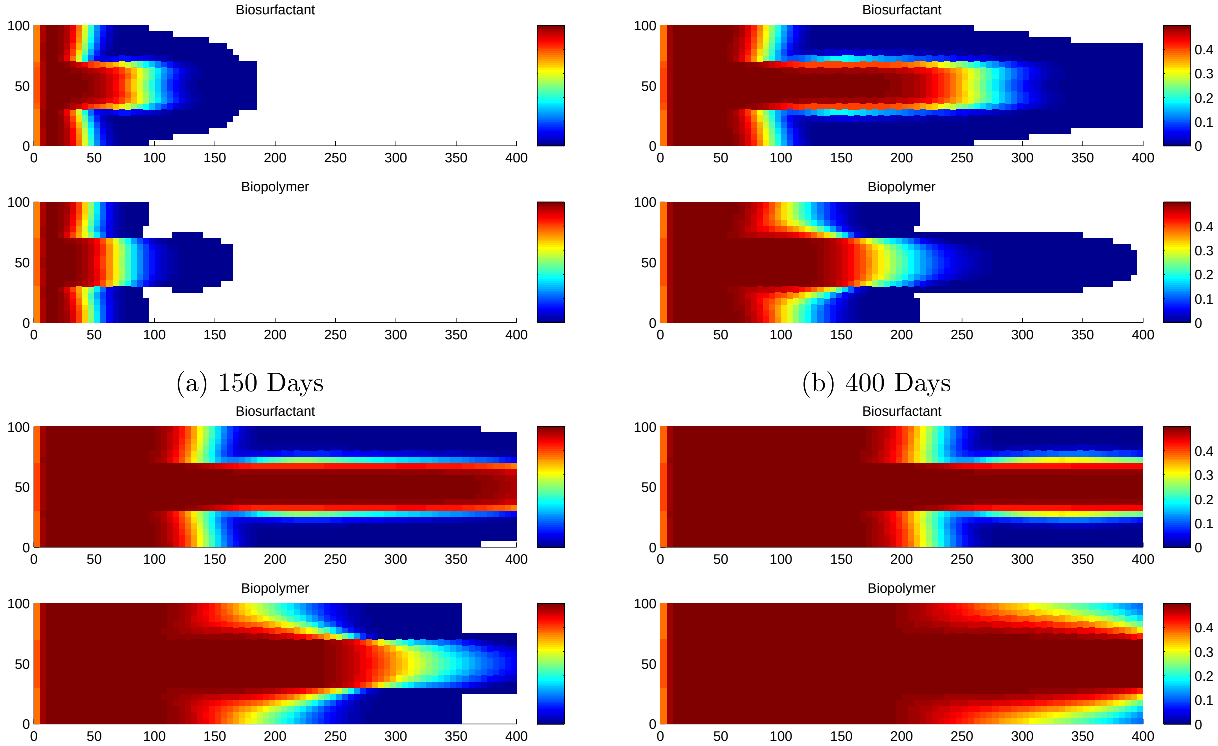
(d) 1250 Days

Figure 7.5: Adsorbed bacteria concentration for the two MEOR thief zone simulations.

residual oil value. This is of course due to the biosurfactant concentrations flowing in the water phase improving the relative permeability. Because the water flows preferentially in the thief zone initially since the absolute permeability is higher, more of the metabolites will flow there and further increase the relative permeability.

7.2 Bacteria and Metabolite Concentration

The adsorption of the bacteria may prove to be a hindrance for the biosurfactant simulation. In Figure 7.5, we see the concentration of adsorbed bacteria in both MEOR simulations at different times. The white area corresponds to there being no bacteria present at all. After 150 days, we can see that the bacteria in the biosurfactant simulation is distributing more in the region outside of the thief zone. It is reasonable to assume that the concentration of bacteria and hence metabolites would be larger in this area if adsorption was not taking place. While the exact patterns of bacteria distribution differ between the two methods, their regions of concentration are quite similar. Obviously the preference lies in the thief zone and near the injection boundary. What is interesting is the similar location of their leading edges. The front through the high and low permeability areas reach the other end of the reservoir in unison after approximately 400 and 750 days respectively. Also the high concentration of bacteria at the near well boundary propagates the same. It is only in the thief zone itself that the bacteria in the biosurfactant simulation sprints ahead slightly. Logically it follows that bacteria concentration influences metabolite concentration. Here there are differences that may at first appear counterintuitive. In Figure 7.6 we have the metabolite concentrations of the two MEOR simulations at different times. The initial profile at 150 days offers no surprises. At 400 days however, the



(a) 150 Days

(b) 400 Days

(c) 700 Days

(d) 1200 Days

Figure 7.6: Metabolite concentration of the two MEOR thief zone simulations at different times.

concentration of biopolymer begins to display a wider distribution in the area of lower permeability. This is clearly visible at 700 and 1200 days as well. The biopolymer is more evenly spread and the biosurfactant is traveling with the sharp profile of a front. If there was no bacterial adsorption, the biopolymer would likely be traveling as a front as well. However, as seen previously in Figure 6.13, the adsorbed bacteria creates a metabolite front with a much shallower slope. The reason this is not seen in the biosurfactant simulation is that the concentration of metabolites is already high enough to change the IFT and the relative permeability to create a water front.

7.3 Evaluation

This test was designed to investigate how a thief zone affects MEOR. The conclusion that can be drawn is that the effect is dependent on the type of metabolite being produced. As expected, the biopolymer found little difficulty in overcoming the thief zone and recovered over 52% OOIP after just 1000 days and continued slowly to 59% after 3000. The thief zone effect was more prevalent in the biosurfactant simulation. Recovery did not occur as quickly and only 43% OOIP was produced after 1000 days. This number also climbed to 59% after 3000 days. Even though they both reach this value at the same time, the biopolymer is the preferred method. If this were a real reservoir, a larger profit could be made by halting production earlier and biopolymer is the most efficient method in that scenario. While this is certainly not a real reservoir, it does serve to highlight how changes in permeability control fluid flow. Adding a third dimension to this simulation will also bring gravitational effects into consideration. This was done with $\Delta z = 20$ m. The results are effectively qualitatively identical though quantitatively, recovery is reduced. Further time should be spent studying the full impacts of three-dimensional flow before any conclusions can be made.

Chapter 8

Conclusion and Improvements

We have developed an MEOR model that combines porous media flow with microbial kinetics. Two types of metabolites are modeled with their different effects on the fluid flow. Bacterial adsorption is also included in the model which can form a biofilm and decrease porosity and permeability. The model was then implemented to be used in conjunction with MRST. We attempted to then recreate results of two MEOR simulations previously published, implementing both biosurfactant and biopolymer as the active metabolites. The goal of the recreations was to verify the validity of the model we had developed. The results were not quantitatively identical since the methods of finding solutions were different. Recovery percentage was slightly less in each of our simulations. Qualitatively, the desired results were achieved. In the recreation of biosurfactant simulations from Nielsen et al. [16], we used equation (4.2) to simulate a reduction of the IFT between the oil and water phases. We then changed the residual oil value, S_{or}, used in the Corey relative permeability calculation. The result was the creation of an oil bank and second water front that greatly increased the recovery. For the simulation of biopolymer, a new water viscosity value is calculated based on the biopolymer concentration. To recreate the results from Lacerda et al. (2012) [7], there was an issue with not knowing all the parameters used in their original simulation. Therefore, comparison of quantitative results is mostly pointless, though the numbers do serve to provide a general expectation of the improvement level. The results qualitatively are in congruence again. Thus, we are confident that both the biopolymer and biosurfactant MEOR models were implemented correctly. Then, both simulations were ran again, this time with Langmuir equilibrium adsorption being applied to the bacteria. This simulated biofilm formation and allowed for metabolites to be produced nearer the injection boundaries. The result was a very slight increase in the recovery. Afterwards, a simulation allowing for two-dimensional flow was ran on a reservoir with a channel of high permeability running through the middle of it. This was meant to simulate a thief zone, structures that are commonly found in real reservoirs and that negatively impact oil recovery. Both MEOR metabolites improve upon the standard water flood though the biopolymer is the only one that truly negates the impact of the thief zone. While these tests were conceptually simple and do not resemble any kind of realistic reservoirs, they do serve an important purpose. They allow for an increased emphasis and focus on various traits encountered in a realistic reservoir. Had there only been a single simulation designed for a synthetic reservoir with realistic structure and petrophysical properties, there would have been too much data to properly analyze. That is not to say that such a simulation is worthless, on the contrary it seems to be more the logical ultimate goal. However, steps must be taken in preparation to first understand the basic behavior of the model. This thesis has attempted to take those first steps. There are still improvements that may be made to prepare for the test of a realistic reservoir. An obvious addition would be successful implementation of the porosity and permeability modifications due to biofilm. Attempts were made at this though the change to the results was either non-existent or unrealistically large. The study of three-dimensional flow should be done to observe the effect of gravity on the distribution of microbes. As they travel in water and the water is denser than the oil, the bacteria and metabolites would not be uniformly distributed vertically. Exactly how this impacts the effectiveness of MEOR should be examined. Another improvement could be the addition of competing, in situ bacteria. Although research into the reservoir is done prior to the use of MEOR, if there exists a competing bacteria that is not discovered, it would be useful to study how that would affect implementation. Also, adsorption could be added for nutrients and biosurfactant instead of only bacteria and biopolymer. Additional bacterial byproducts such as gases and acids are also of interest and could be included. It is also advisable to perform a small scale simulation of a reservoir with realistic petrophysical parameters to identify any weaknesses in the model before an attempt is made at one of the realistic synthetic reservoirs available for testing upon. As mentioned before, it would be easy to overlook a problem if there is not a concerted effort to discover it.

Overall the work of this thesis has striven to create a functioning, utilitarian MEOR simulation model which may be implemented with a variety of metabolite effects and microbial mechanisms. This has been evaluated through the variety of tests presented in this work. While certain benchmark tests have not yet been completed, to this point, the simulation model has successfully behaved as expected. The bacteria consumes nutrients to multiply and produce metabolites, the metabolites alter the fluid properties correctly, and the end result is an increase in the amount of recovered oil. While there are improvements that can be made, the most important aspects of the MEOR simulation model are present and functioning properly. For that reason, it is our assessment that the goal has been reached successfully.

Appendix A

List of Symbols and Abbreviations

Symbol	**Unit**	**Description**
A_v	m^2/m^3	Ratio of surface area to volume
a	-	Exponent for IFT interpolation function
B	kg/m^3	Concentration of bacteria
B_a	kg/m^2	Concentration of adsorbed bacteria
B_f	kg/m^3	Concentration of bacteria in water phase
c	kg/m^3	Concentration of generic substance
$\vec{e}_z$	-	Basis vector for vertical axis
i	-	Generic phase index
$\mathbf{K}$	mD	Absolute permeability tensor
$\mathcal{K}$	-	Surfactant partitioning constant
kr_o	-	Relative permeability of oil
kr_w	-	Relative permeability of water
kr_{owi}	-	Relative permeability value of oil at initial water saturation
kr_{wor}	-	Relative permeability value of water at residual oil saturation
$kr_{i,base}$	-	Base relative permeability for Coats' interpolation
$kr_{i,misc}$	-	Miscible relative permeability for Coats' interpolation
$l_{\{1,2,3\}}$	-	Parameters for biosurfactant efficacy
M	kg/m^3	Concentration of generic metabolite
M_s	kg/m^3	Concentration of biosurfactant metabolite
M_{so}	kg/m^3	Partition of biosurfactant in oil
M_{sw}	kg/m^3	Partition of biosurfactant in water
M_p	kg/m^3	Concentration of biopolymer metabolite
$M_{p,a}$	kg/m^3	Concentration of adsorbed biopolymer
$M_{p,\max}$	kg/m^3	Maximum effective concentration of biopolymer
$\bar{M}_p$	-	Ratio of $M_p/M_{p,\max}$
N	kg/m^3	Concentration of nutrient
N_{crit}	kg/m^3	Critical nutrient concentration value for metabolite production
N_{ca}	-	Capillary number, relates viscosity to capillary forces
n	-	Exponent in Corey relative permeability equation
o	-	Subscript to indicate oil phase
p	Pa	Pressure
q_b	-	Source term in conservation equations for bacteria concentration
q_n	kg/m^3	Source term in conservation equations for nutrient concentration

q_o	-	Source term in conservation equations for oil
q_w	m^3	Source term in conservation equations for water
$\tilde{q}_i$	-	Source term altered for fluid compressibility
R_b	-	Reaction term of bacteria in conservation equations
R_m	-	Reaction term of metabolite in conservation equations
R_n	-	Reaction term of nutrient in conservation equations
$R(M_{p,a})$	-	Retardation function due to adsorbed biopolymer
S_o	-	Saturation ratio of oil
S_w	-	Saturation ratio of water
S_{or}	-	Residual oil saturation
S_{wi}	-	Initial water saturation
$\mathcal{S}$	m^2	Available surface area for adsorption
$\vec{u}_o$	m/s	Darcy velocity of oil
$\vec{u}_w$	m/s	Darcy velocity of water
V	m^3	Elementary volume
Y_b	-	Yield of bacteria from Monod model
Y_m	-	Yield of metabolite from Monod model
β_o	-	Formation volume factor of oil
β_w	-	Formation volume factor of water
η_o	cP	Viscosity of oil
η_w	cP	Viscosity of water
μ_b	day^{-1}	Monod model growth rate of bacteria
$\mu_{b,max}$	day^{-1}	Monod model maximum growth rate of bacteria
μ_m	day^{-1}	Monod model production rate of metabolite
$\mu_{m,max}$	day^{-1}	Monod model maximum production rate of metabolite
ρ_o	kg/m^3	Density of oil
$\rho_{sc,o}$	kg/m^3	Surface density of oil
$\rho_{sc,w}$	kg/m^3	Surface density of water
ρ_w	kg/m^3	Density of water
σ	mN/m	Interfacial tension
τ	-	Tortuosity
ϕ	-	Porosity
χ	-	Parameter placeholder for Corey interpolation function
ψ	-	Biofilm
$\omega_{\{1,2\}}$	kg/m^2, m^3/kg	Langmuir distribution parameters

Subscripts and superscripts	**Meaning**
$*$	Superscript used when calculating a new property value
i	Subscript used to refer to generic phase
j	Subscript to refer to surfactant properties in equation (4.2)
0	Subscript to refer to initial value of a property

Abbreviation	**Name**
ASP	Alkaline surfactant polymer
EOR	Enhanced oil recovery
EPS	Extracellular polymeric substances
IFT	Interfacial tension

MEOR	Microbial enhanced oil recovery
MRST	MATLAB Reservoir Simulation Toolbox
REV	Representative elementary volume
PVI	Pore volume injection

Appendix B

MATLAB code

B.1 Model

This is the model file that inherits from TwoPhaseOilWaterModel. To be used properly with MRST, this needs to be placed in the following directory .../ mrst-autodiff / ad-blackoil / models.

```
classdef MEORaModel < TwoPhaseOilWaterModel
    % Oil/water/microbial system
    % This model is a two phase oil/water model, extended with the
    % microbial phase in addition.
    % Microbe effects currently available for simulation:
    % Biopolymer, Biosurfactant, Bacterial Adsorption

    properties
        % Substances in reservoir
        microbe
        nutrient
        metabolite
        biofilm
        stateplots
        % Variables
        yield_microbe
        yield_metabolite
        growth_max_microbe
        growth_max_metabolite
        halfsat_microbe
        halfsat_metabolite
        crit_val
        biosurf
        biopoly
        langmuir
    end

    methods
        function model = MEORaModel(G, rock, fluid, varargin)
            model = model@TwoPhaseOilWaterModel(G,rock, fluid);
            model.microbe = true;
            model.metabolite = true;
            model.nutrient = true;
            % pre-defining defaults, can be overwritten
            model.yield_microbe = .5;
            model.yield_metabolite = .5;
        model.growth_max_microbe = .2/day;
        model.growth_max_metabolite = .2/day;
        model.halfsat_microbe = .5;
        model.halfsat_metabolite = .5;
        model.crit_val = 0;
        model.langmuir = [0, 0];

        % type of metabolite to be defined manually
        model.biosurf = false;
        model.biopoly = false;
        model.biofilm = false;
        modle.stateplots = false;
        % not available for more than 3 phases
        % oil/water/microbe/nutrient/metabolite
        model.outputFluxes = true;
        model.wellVarNames = {'qWs', 'qOs', 'qWMEOR', 'bhp'};
        model = merge_options(model, varargin{:});
    end


    function [problem, state] = getEquations(model, state0, state,...
            dt, drivingForces, varargin)
        if model.biofilm
        [problem, state] = equationsMEORbiofilm(state0, state, model,...
            dt, drivingForces, varargin{:});
        else
        [problem, state] = equationsMEORa(state0, state, model,...
            dt, drivingForces, varargin{:});
        end
    end

    function [fn, index] = getVariableField(model, name)
        switch(lower(name))
            case 'microbe'
                fn = 'm';
                index = 1;
            case 'nutrient'
                fn = 'n';
                index = 1;
            case 'metabolite'
                fn = 'meta';
                index = 1;
            case 'biofilm'
                fn = 'bio';
                index = 1;
            otherwise
                [fn, index] = getVariableField@TwoPhaseOilWaterModel(model, name);
        end
    end

    function [state, report] = updateState(model, state, problem, ...
            dx, drivingForces)
       [state, report] = updateState@TwoPhaseOilWaterModel(model, ...
           state, problem, dx, drivingForces);
       if model.stateplots && rem(problem.iterationNo,5)==1
           xvals = linspace(0,1,model.G.cells.num);
           set(0, 'currentfigure', 1);
           plot(xvals, state.pressure);
           title(sprintf('Pressure'));
```

```
                set(0, 'currentfigure', 2);
                plot(xvals, state.s);
                title(sprintf('Saturation'));
                set(0, 'currentfigure', 3);
                plot(xvals, state.m);
                title(sprintf('Microbe concentration'));
                set(0, 'currentfigure', 4);
                plot(xvals, state.n);
                title(sprintf('Nutrient concentration'));
                set(0, 'currentfigure', 5);
                plot(xvals, state.meta);
                title(sprintf('Metabolite concentration'));
                if model.biofilm
                set(0, 'currentfigure', 6);
                plot(xvals, state.bio);
                title(sprintf('biofilm concentration'));
                end
                drawnow;
            end
        end
    end
end
```

B.2 Equations files

There are two different equations files. One for use with biofilm and one without biofilm functionality. These should be placed in the directory .../ mrst-autodiff / ad-blackoil / utils.

```
function [problem, state] =
    equationsMEORbiofilm(state0,state,model,dt,drivingForces,varargin)
% Work in progress to create MEOR effects
% Get linearized problem for oil/water/MEOR system with black oil
% properties
opt = struct('Verbose', mrstVerbose,...
             'reverseMode', false,...
             'resOnly', false,...
             'iteration', -1);

opt = merge_options(opt, varargin{:});

W = drivingForces.Wells;

% Operators, grid, and fluid model
s = model.operators;
G = model.G;
f = model.fluid;
Y_micro = model.yield_microbe;
Y_meta = model.yield_metabolite;
mu_micro = model.growth_max_microbe;
mu_meta = model.growth_max_metabolite;
K_micro = model.halfsat_microbe;
K_meta = model.halfsat_metabolite;
N = model.crit_val;
w = model.langmuir;

% Properties at current timestep
```

```
[p, sW, m, meta, n, bio, wellSol] = model.getProps(state, 'pressure', 'water',...
    'microbe', 'metabolite', 'nutrient','biofilm', 'wellsol');

% Properties at previous timestep
[p0,sW0, m0, meta0, n0,bio0] = model.getProps(state0, 'pressure', 'water', ...
    'microbe', 'metabolite', 'nutrient','biofilm');

pBH = vertcat(wellSol.bhp);
qWs = vertcat(wellSol.qWs);
qOs = vertcat(wellSol.qOs);
qWMEOR = vertcat(wellSol.qWMEOR);

% Initialize independent variables
if ~opt.resOnly,
    % ADI variables needed since we are not only computing residuals.
    if ~opt.reverseMode,
        [p, sW, m, meta, n,bio, qWs, qOs, qWMEOR, pBH] = ...
            initVariablesADI( p, sW, m, meta, n,bio, qWs, qOs, qWMEOR, pBH);
    else
        [p0, sW0, m0, meta0, n0,bio0, tmp,tmp,tmp,tmp] = ...
            initVariablesADI(p0, sW0, m0, meta0, n0,bio0,...
            zeros(size(qWs)), zeros(size(qOs)), zeros(size(qWMEOR)),...
            zeros(size(pBH)));
        clear tmp
    end
end

% We will solve for pressure, water saturation (oil saturation follows from
% the definition of saturations) ((may need to change later)), microbe
% concentration, nutrient concentration, metabolite concentration,
% and well rates and bhp.
primaryVars = {'pressure', 'sW', 'microbe', 'metabolite', 'nutrient','biofilm', 'qWs', 'qOs',...
    'qWMEOR', 'bhp'};

% Evaluate relative permeability
sO = 1 - sW;
sO0 = 1 - sW0;

% Find effective surface area for adsorption
SurfA = 3.*10^5*sW./model.rock.poro;
[krW, krO] = model.evaluteRelPerm({sW, sO}); % as written

if model.biosurf
    % form is taken from Nielsen
    % constants must be changed for new rel perm curve
    partition = 1.*(sW.*model.fluid.rhoWS)./(sO.*model.fluid.rhoOS);
    meta_eff = meta.*partition./(partition + 1);
    surfa = [41*10^-4, 2, 180];   % Surfactant efficacy defined here
    ift = @(s) 29.*(-tanh(surfa(3).*s-surfa(2))+1+surfa(1))./...
        (-tanh(-surfa(2))+1+surfa(1));
    sigma = ift(double(meta_eff));
    f = (sigma/29).^(1/6);
    sor = f.*.23;               % Only defined for a specific
    f = ones(length(f),1);      % Corey equation in use
    wmax = f.*.3+1-f;           % Also only currently changing
    swi = f.*.16;               % resdiual oil.
    omax = f.*.7+1-f;
    a = f.*2+1-f;
    krO_max = omax.*((sO-sor)./(1-swi-sor)).^a;
    krW_max = wmax.*((sW-swi)./(1-swi-sor)).^a;
    inx = meta_eff>1e-16;
    krW = krW + (krW_max - krW).*inx;
    krO = krO + (krO_max - krO).*inx;
end

% Multipliers for properties
[pvMult, transMult, mobMult, pvMult0] = getMultipliers(model.fluid, p,p0);


% Modify relperm by mobility multiplier
krW = mobMult.*krW;
krO = mobMult.*krO;

% Adjustments for pore volume and relperm for biofilm
if model.biofilm
    bioeff = bio;
    bioeff(bio<0) = 0;
    psi = bioeff./1000; % number is biofilm density
    phi_rel = 1 - psi;
  % pvMult = pvMult.*phi_rel;
  % krW = krW.*(phi_rel.^(19/6)); % plenty of other ways found in Thullner
end

% Compute transmissibility
T = s.T.*transMult;

% Gravity contribution
gdz = model.getGravityGradient();

% Evaluate water and MEOR props
[vW, vMicro, vMeta, vN, bW, mobW, mobM_0, mobPol, mobN, rhoW, pW, upcw] = ...
    getFluxAndPropsMEORa(model, p, sW, m, meta, n, krW, T, gdz);
bW0 = model.fluid.bW(p0);

% Evaluate Oil properties
[vO,bO,mobO,rhoO,p,upco] = getFluxAndPropsOil_BO(model,p,sO,krO,T,gdz);
bO0 = getbO_BO(model, p0);

if model.outputFluxes
    state = model.storeFluxes(state, vW, vO, vMicro);
end

if model.extraStateOutput
    state = model.storebfactors(state, bW, bO, []);
    state = model.storeMobilities(state, mobW, mobO, mobM_0, mobN);
    state = model.storeUpstreamIndices(state, upcw, upco, []);
end


% EQUATIONS ---------------------------------------------------------------
% Microbe, biofilm, and metabolite calculations
mu_b = mu_micro.*n./(K_micro + n);
mu_m = mu_meta.*(n-N)./(K_meta + n - N);
R_n = -mu_b.*(m.*sW.*bW+bio).*Y_micro - mu_m.*(m.*sW.*bW+bio).*Y_meta;
lang = w(1).*w(2).*(m-bio)./(1+w(2).*(m-bio));

bWvW = s.faceUpstr(upcw, bW).*vW;
```

```
bWvMicro = s.faceUpstr(upcw, bW).*vMicro;
bWvMeta = s.faceUpstr(upcw, bW).*vMeta;
bWvN = s.faceUpstr(upcw, bW).*vN;
bOvO = s.faceUpstr(upco, bO).*vO;


% Conservation of oil:
oil = (s.pv/dt).*(pvMult.*bO.*sO - pvMult0.*bO0.*sO0) + s.Div(bOvO);

%Conservation of  water:
water = (s.pv/dt).*(pvMult.*bW.*sW - pvMult0.*bW0.*sW0) + s.Div(bWvW);

%Conservation of  microbes:
microbe = (s.pv/dt).*((pvMult.*bW.*sW.*m - pvMult0.*bW0.*sW0.*m0)) + ...
              s.pv.*pvMult.*SurfA.*lang - ...
              s.pv.*mu_b.*(m.*sW.*bW+bio).*pvMult.*Y_micro + s.Div(bWvMicro);
biofilm = (s.pv/dt).*((pvMult.*bio - pvMult0.*bio0)) - s.pv.*pvMult.*SurfA.*lang;

%Conservation of  nutrients:
nutrient = (s.pv/dt).*((pvMult.*bW.*sW.*n - pvMult0.*bW0.*sW0.*n0))  -...
              s.pv.*R_n.*pvMult+ s.Div(bWvN);

%Conservation of metabolites:
metabolite = (s.pv/dt).*((pvMult.*bW.*sW.*meta - pvMult0.*bW0.*sW0.*meta0))...
              - s.pv.*mu_m.*(m.*sW.*bW+bio).*pvMult.*Y_meta+ s.Div(bWvMeta);

eqs = {water, oil, microbe, nutrient, metabolite, biofilm};
names = {'water', 'oil', 'microbe', 'nutrient', 'metabolite', 'biofilm'};
types = {'cell', 'cell', 'cell', 'cell', 'cell', 'cell'};

% Add in any fluxes/source terms given as boundary conditions
[eqs, qBC, BCTocellMap, qSRC, srcCells] = addFluxesFromSourcesAndBC(...
    model, eqs, {pW, p}, {rhoW, rhoO}, {mobW, mobO}, {bW, bO}, ...
    {sW, sO}, drivingForces);

% Add MEOR boundary conditions
if ~isempty(drivingForces.bc) && isfield(drivingForces.bc,'m')
    injInx = qBC{1} > 0; % Water inflow indicies
    mbc = (BCTocellMap')*m; % m_0 is only type injected
    nbc = (BCTocellMap')*n;
    metabc = (BCTocellMap')*meta;
    mbc(injInx) = drivingForces.bc.m(injInx);
    nbc(injInx) = drivingForces.bc.n(injInx);
    metabc(injInx) = drivingForces.bc.meta(injInx);
    eqs{3} = eqs{3} - BCTocellMap*(mbc.*qBC{1});
    eqs{4} = eqs{4} - BCTocellMap*(nbc.*qBC{1});
    eqs{5} = eqs{5} - BCTocellMap*(metabc.*qBC{1});
end

% Add MEOR source
if ~isempty(drivingForces.src) && isfield(drivingForces.src, 'm')
    injInx = qSRC{1}>0;
    msrc = m(srcCells);
    nsrc = n(srcCells);
    metasrc = meta(srcCells);
    msrc(injInx) = drivingForces.src.m(injInx);
    nsrc(injInx) = drivingForces.src.n(injInx);
    eqs{3}(srcCells) = eqs{3}(srcCells) - msrc.*qSRC{1};
    eqs{4}(srcCells) = eqs{4}(srcCells) - nsrc.*qSRC{1};
```

```
 %   eqs{5}(srcCells) = eqs{5}(srcCells) - metasrc.*qSRC{1};
end

% WELLS NOT READY FOR BIOFILM IMPLEMENTATION
 if ~isempty(W)
    wm = model.wellmodel;
    if ~opt.reverseMode
        wc = vertcat(W.cells);
        pw = p(wc);
        rhos = [f.rhoWS, f.rhoOS];
        bw = {bW(wc), bO(wc)};
        tw = {mobW(wc), mobO(wc)};
        s = {sW(wc), sO(wc)};
        [cqs, weqs, ctrleqs, wc, state.wellSol] = ...
            wm.computeWellFlux(model, W, wellSol,...
            pBH, {qWs, qOs}, pw, rhos, bw, tw, s, {},...
            'nonlinearIteration', opt.iteration);
        % Store the well equations (relating well BHP to influx)
        eqs(6:7) = weqs;
        % Store control equations
        eqs{9} = ctrleqs;
        % Add source terms to the equations.
        eqs{1}(wc) = eqs{1}(wc) - cqs{1};
        eqs{2}(wc) = eqs{1}(wc) - cqs{2};
        % MEOR well equations
         [~, wciMEOR, iInxW, MEORc] = getWellMEOR(W);
         mw = m(wc);
         nw = n(wc);
         mw(iInxW) = wciMEOR.*(1-MEORc);
         nw(iInxW) = wciMEOR.*MEORc;

        bWqM = mw.*cqs{1};
        bWqN = nw.*cqs{1};
        eqs{3}(wc) = eqs{3}(wc) - bWqM;
        eqs{4}(wc) = eqs{5}(wc) - bWqN;

        % Well MEOR rate for each well is water rate in each perforation
        % multiplied with microbe and nutrient concentration in that
        % perforated cell
        perf2well = getPerforationToWellMapping(W);
        Rw = sparse(perf2well, (1:numel(perf2well))', 1,numel(W),numel(perf2well));
        eqs{8} = qWMEOR - Rw*(cqs{1}.*(mw+nw));
        names(6:9) = {'waterWells', 'oilWells', 'meorWells', 'closureWells'};
        types(6:9) = {'perf', 'perf', 'perf', 'well'};
    else
        [eq, n, typ] = ...
            wm.createReverseModeWellEquations(model, state0.wellSol, p0);
        % add another equation for MEOR well rates.
        [eqs{6:9}] = deal(eq{1});
        [names{6:9}] = deal(n{1});
        [types{6:9}] = deal(typ{1});
    end
end
problem = LinearizedProblem(eqs,types,names,primaryVars,state,dt);
end

function [wMEOR, wciMEOR, iInxW,MEORc] = getWellMEOR(W)
    if isempty(W)
        wMEOR = [];
        wciMEOR = [];
        iInxW = [];
        MEORc = [];
        return
    end
    inj = vertcat(W.sign) == 1;
    mInj = cellfun(@(x)~isempty(x), {W(inj).meor});
    wMEOR = zeros(nnz(inj),1);
    wMEOR(mInj) = vertcat(W(inj(mInj)).meor);
    wciMEOR = rldecode(wMEOR, cellfun(@numel, {W(inj).cells}));
    MEORcomp = zeros(nnz(inj),1);
    MEORcomp(mInj) = [.5];
    MEORc = rldecode(MEORcomp, cellfun(@numel, {W(inj).cells}));

    % Injection cells
    nPerf = cellfun(@numel, {W.cells})';
    nw = numel(W);
    perf2well = rldecode((1:nw)',nPerf);
    compi = vertcat(W.compi);
    iInx = rldecode(inj, nPerf);
    iInx = find(iInx);
    iInxW = iInx(compi(perf2well(iInx),1)==1);
end
```

```
function [problem, state] = equationsMEORa(state0,state,model,dt,...
                                           drivingForces,varargin)
% Get linearized problem for oil/water/MEOR system with black oil
% properties
opt = struct('Verbose', mrstVerbose,...
             'reverseMode', false,...
             'resOnly', false,...
             'iteration', -1);

opt = merge_options(opt, varargin{:});

W = drivingForces.Wells;

% Operators, grid, and fluid model
s = model.operators;
G = model.G;
f = model.fluid;
Y_micro = model.yield_microbe;
Y_meta = model.yield_metabolite;
mu_micro = model.growth_max_microbe;
mu_meta = model.growth_max_metabolite;
K_micro = model.halfsat_microbe;
K_meta = model.halfsat_metabolite;
N = model.crit_val;


% Properties at current timestep
[p, sW, m, meta, n, wellSol] = model.getProps(state, 'pressure', 'water',...
    'microbe', 'metabolite', 'nutrient', 'wellsol');

% Properties at previous timestep
[p0,sW0, m0, meta0, n0] = model.getProps(state0, 'pressure', 'water', ...
    'microbe', 'metabolite', 'nutrient');

```

```
pBH = vertcat(wellSol.bhp);
qWs = vertcat(wellSol.qWs);
qOs = vertcat(wellSol.qOs);
qWMEOR = vertcat(wellSol.qWMEOR);

% Initialize independent variables
if ~opt.resOnly,
    % ADI variables needed since we are not only computing residuals.
    if ~opt.reverseMode,
        [p, sW, m, meta, n, qWs, qOs, qWMEOR, pBH] = ...
            initVariablesADI( p, sW, m, meta, n, qWs, qOs, qWMEOR, pBH);
    else
        [p0, sW0, m0, meta0, n0, tmp,tmp,tmp,tmp] = ...
            initVariablesADI(p0, sW0, m0, meta0, n0,...
            zeros(size(qWs)), zeros(size(qOs)), zeros(size(qWMEOR)),...
            zeros(size(pBH)));
        clear tmp
    end
end

% We will solve for pressure, water saturation (oil saturation follows from
% the definition of saturations), microbe
% concentration, nutrient concentration, metabolite concentration,
% and well rates and bhp.
primaryVars = {'pressure', 'sW', 'microbe', 'metabolite', 'nutrient',...
                'qWs', 'qOs', 'qWMEOR', 'bhp'};

% Evaluate relative permeability
sO = 1 - sW;
sO0 = 1 - sW0;

[krW, krO] = model.evaluteRelPerm({sW, sO}); % as written

if model.biosurf
    % form is taken from Nielsen
    % constants are as well
    partition = 1.*(sW.*model.fluid.rhoWS)./(sO.*model.fluid.rhoOS);
    meta_eff = meta.*partition./(partition + 1);
    surfa = [1*10^-4, 0.2, 1.5*10^4]; % Surfactant efficacy defined here
    ift = @(s) 29.*(-tanh(surfa(3).*s-surfa(2))+1+surfa(1))./...
                (-tanh(-surfa(2))+1+surfa(1));
    sigma = ift(double(meta_eff));
    f = (sigma/29).^(1/6);
    sor = f.*.4;                % Only defined for a specific
    f = ones(length(f),1);      % Corey equation in use
    wmax = f.*.5+1-f;           % Also only currently changing
    swi = f.*.3;                % residual oil.
    omax = f.*.8+1-f;
    a = f.*2+1-f;
    krO_max = omax.*((sO-sor)./(1-swi-sor)).^a;
    krW_max = wmax.*((sW-swi)./(1-swi-sor)).^a;
    inx = meta_eff>1e-16;
    krW = krW + (krW_max - krW).*inx;
    krO = krO + (krO_max - krO).*inx;
end

% Multipliers for properties
[pvMult, transMult, mobMult, pvMult0] = getMultipliers(model.fluid, p,p0);

```

```
% Modify relperm by mobility multiplier
krW = mobMult.*krW;
krO = mobMult.*krO;

% Compute transmissibility
T = s.T.*transMult;

% Gravity contribution
gdz = model.getGravityGradient();

% Evaluate water and MEOR props
[vW, vMicro, vMeta, vN, bW, mobW, mobM_0, mobPol, mobN, rhoW, pW, upcw] = ...
    getFluxAndPropsMEORa(model, p, sW, m, meta, n, krW, T, gdz);
bW0 = model.fluid.bW(p0);

% Evaluate Oil properties
[vO,bO,mobO,rhoO,p,upco] = getFluxAndPropsOil_BO(model,p,sO,krO,T,gdz);
bO0 = getbO_BO(model, p0);

if model.outputFluxes
    state = model.storeFluxes(state, vW, vO, vMicro);
end

if model.extraStateOutput
    state = model.storebfactors(state, bW, bO, []);
    state = model.storeMobilities(state, mobW, mobO, mobM_0, mobN);
    state = model.storeUpstreamIndices(state, upcw, upco, []);
end


% EQUATIONS ---------------------------------------------------------------
% Microbe and metabolite calculations
mu_b = mu_micro.*n./(K_micro + n);
mu_m = mu_meta.*(n-N)./(K_meta + n - N);
R_n = -mu_b.*m.*Y_micro - mu_m.*m.*Y_meta;

bWvW = s.faceUpstr(upcw, bW).*vW;
bWvMicro = s.faceUpstr(upcw, bW).*vMicro;
bWvMeta = s.faceUpstr(upcw, bW).*vMeta;
bWvN = s.faceUpstr(upcw, bW).*vN;
bOvO = s.faceUpstr(upco, bO).*vO;

% Conservation of oil:
oil = (s.pv/dt).*(pvMult.*bO.*sO - pvMult0.*bO0.*sO0) + s.Div(bOvO);

%Conservation of  water:
water = (s.pv/dt).*(pvMult.*bW.*sW - pvMult0.*bW0.*sW0) + s.Div(bWvW);

%Conservation of  microbes:
microbe = (s.pv/dt).*((pvMult.*bW.*sW.*m - pvMult0.*bW0.*sW0.*m0))  -...
            s.pv.*mu_b.*m.*bW.*sW.*pvMult.*Y_micro + s.Div(bWvMicro);

%Conservation of  nutrients:
nutrient = (s.pv/dt).*((pvMult.*bW.*sW.*n - pvMult0.*bW0.*sW0.*n0)) - ...
            s.pv.*R_n.*bW.*sW.*pvMult+ s.Div(bWvN);

%Conservation of metabolites:
metabolite =(s.pv/dt).*((pvMult.*bW.*sW.*meta - pvMult0.*bW0.*sW0.*meta0))...
                - s.pv.*mu_m.*m.*bW.*sW.*pvMult.*Y_meta+ s.Div(bWvMeta);
```

```matlab

eqs = {water, oil, microbe, nutrient, metabolite};
names = {'water', 'oil', 'microbe', 'nutrient', 'metabolite'};
types = {'cell', 'cell', 'cell', 'cell', 'cell'};

% Add in any fluxes/source terms given as boundary conditions
[eqs, qBC, BCTocellMap, qSRC, srcCells] = addFluxesFromSourcesAndBC(...
    model, eqs, {pW, p}, {rhoW, rhoO}, {mobW, mobO}, {bW, bO}, ...
    {sW, sO}, drivingForces);

% Add MEOR boundary conditions
if ~isempty(drivingForces.bc) && isfield(drivingForces.bc,'m')
    injInx = qBC{1} > 0; % Water inflow indicies
    mbc = (BCTocellMap')*m;
    nbc = (BCTocellMap')*n;
    metabc = (BCTocellMap')*meta;
    mbc(injInx) = drivingForces.bc.m(injInx);
    nbc(injInx) = drivingForces.bc.n(injInx);
    metabc(injInx) = drivingForces.bc.meta(injInx);
    eqs{3} = eqs{3} - BCTocellMap*(mbc.*qBC{1});
    eqs{4} = eqs{4} - BCTocellMap*(nbc.*qBC{1});
    eqs{5} = eqs{5} - BCTocellMap*(metabc.*qBC{1});
end

% Add MEOR source
if ~isempty(drivingForces.src) && isfield(drivingForces.src, 'm')
    injInx = qSRC{1}>0;
    msrc = m(srcCells);
    nsrc = n(srcCells);
    metasrc = meta(srcCells);
    msrc(injInx) = drivingForces.src.m(injInx);
    nsrc(injInx) = drivingForces.src.n(injInx);
    eqs{3}(srcCells) = eqs{3}(srcCells) - msrc.*qSRC{1};
    eqs{4}(srcCells) = eqs{4}(srcCells) - nsrc.*qSRC{1};
    eqs{5}(srcCells) = eqs{5}(srcCells) - metasrc.*qSRC{1};
end

% well equations
if ~isempty(W)
    wm = model.wellmodel;
    if ~opt.reverseMode
        wc = vertcat(W.cells);
        pw = p(wc);
        rhos = [f.rhoWS, f.rhoOS];
        bw = {bW(wc), bO(wc)};
        tw = {mobW(wc), mobO(wc)};
        s = {sW(wc), sO(wc)};
        [cqs, weqs, ctrleqs, wc, state.wellSol] = ...
            wm.computeWellFlux(model, W, wellSol,...
            pBH, {qWs, qOs}, pw, rhos, bw, tw, s, {},...
             'nonlinearIteration', opt.iteration);

        % Store the well equations (relating well BHP to influx)
        eqs(6:7) = weqs;

        % Store control equations
        eqs{9} = ctrleqs;
        % Add source terms to the equations.
        eqs{1}(wc) = eqs{1}(wc) - cqs{1};
        eqs{2}(wc) = eqs{1}(wc) - cqs{2};

        % MEOR well equations
         [~, wciMEOR, iInxW, MEORc] = getWellMEOR(W);
         mw = m(wc);
         nw = n(wc);
         mw(iInxW) = wciMEOR.*(1-MEORc);
         nw(iInxW) = wciMEOR.*MEORc;
         bWqM = mw.*cqs{1};
         bWqN = nw.*cqs{1};
         eqs{3}(wc) = eqs{3}(wc) - bWqM;
         eqs{4}(wc) = eqs{5}(wc) - bWqN;

        % Well MEOR rate for each well is water rate in each perforation
        % multiplied with microbe and nutrient concentration in that
        % perforated cell
        perf2well = getPerforationToWellMapping(W);
        Rw = sparse(perf2well, (1:numel(perf2well))', 1,numel(W),numel(perf2well));
        eqs{8} = qWMEOR - Rw*(cqs{1}.*(mw+nw));
        names(6:9) = {'waterWells', 'oilWells', 'meorWells', 'closureWells'};
        types(6:9) = {'perf', 'perf', 'perf', 'well'};
    else
        [eq, n, typ] = ...
            wm.createReverseModeWellEquations(model, state0.wellSol, p0);
        [eqs{6:9}] = deal(eq{1});
        [names{6:9}] = deal(n{1});
        [types{6:9}] = deal(typ{1});
    end
end
problem = LinearizedProblem(eqs,types,names,primaryVars,state,dt);
end

function [wMEOR, wciMEOR, iInxW,MEORc] = getWellMEOR(W)
    if isempty(W)
        wMEOR = [];
        wciMEOR = [];
        iInxW = [];
        MEORc = [];
        return
    end
    inj = vertcat(W.sign) == 1;
    mInj = cellfun(@(x)~isempty(x), {W(inj).meor});
    wMEOR = zeros(nnz(inj),1);
    wMEOR(mInj) = vertcat(W(inj(mInj)).meor);
    wciMEOR = rldecode(wMEOR, cellfun(@numel, {W(inj).cells}));
    MEORcomp = zeros(nnz(inj),1);
    MEORcomp(mInj) = [.5]; % For composition of well MEOR injection
    MEORc = rldecode(MEORcomp, cellfun(@numel, {W(inj).cells}));

    % Injection cells
    nPerf = cellfun(@numel, {W.cells})';
    nw = numel(W);
    perf2well = rldecode((1:nw)',nPerf);
    compi = vertcat(W.compi);
    iInx = rldecode(inj, nPerf);
    iInx = find(iInx);
    iInxW = iInx(compi(perf2well(iInx),1)==1);
end
```

B.2.1 Helper Function for Equations

This is a function used by both equations files. This is also where the biopolymer effects are added. It should also be placed in the directory .../ mrst-autodiff / ad-blackoil / utils.

```
function [vW, vMicro, vMeta, vN, bW, mobW, mobm, mobMeta, mobN, rhoW, pW, upcw]= ...
            getFluxAndPropsMEORa(model, pO, sW, m, meta, n, krW, T, gdz)
fluid = model.fluid;
s = model.operators;

% Check for capillary pressure (p_cOW)
pcOW = 0;
if isfield(fluid, 'pcOW') && ~isempty(sW)
    pcOW = fluid.pcOW(sW);
end
pW = pO - pcOW;

change = 1.*ones(length(meta),1); %0 for power law 1 for parabolic
if model.biopoly
    % Threshold for metabolite effect/hack to avoid complex values
    inx = meta<1e-16;
    meta_eff = meta;
    meta_eff(inx) = 0;
    % power law from Lacerda
    %change = 1.4019.*meta_eff.^.1653;
    % Parabolic law from Bartelds
    change = ((5.*meta_eff).^2 + 5.*meta_eff + 1);
    inx = change<1e-16;
    change(inx) = 0;
end

bW = fluid.bW(pO);
rhoW = bW.*fluid.rhoWS;
% rhoW on face is the average of the neighboring cells
rhoWf = s.faceAvg(rhoW);
muW = fluid.muW(pO);
%muWeff = muW + change.*1e-3; % For power law
muWeff = muW.*change;          % For parabolic law
mobW = krW./muWeff;
dpW = s.Grad(pO-pcOW) - rhoWf.*gdz;
% water upstream index
upcw = double(dpW)<=0;
vW = -s.faceUpstr(upcw,mobW).*T.*dpW;
if any(bW <0)
    warning('Negative water compressibility present')
end

% MEOR
mobm = mobW.*m;
mobMeta = mobW.*meta;
mobN = mobW.*n;
vMicro = -s.faceUpstr(upcw, mobm).*T.*dpW;
vMeta = -s.faceUpstr(upcw, mobMeta).*T.*dpW;
vN = -s.faceUpstr(upcw, mobN).*T.*dpW;
```

Bibliography

[1] Tarek Ahmed. *Working Guide to Reservoir Rock Properties and Fluid Flow.* Elsevier Science, [S.l.], 2009.

[2] In Young Bae, Im-Kyung Oh, Suyong Lee, Sang-Ho Yoo, and Hyeon Gyu Lee. Rheological characterization of levan polysaccharides from microbacterium laevaniformans. *International Journal of Biological Macromolecules*, 42(1):10 – 13, 2008.

[3] G.A. Bartelds, J. Bruining, and J. Molenaar. The modeling of velocity enhancement in polymer flooding. *Transport in Porous Media*, 26(1):75–88, 1997.

[4] John R Fanchi. *Principles of applied reservoir simulation.* Gulf Professional Publishing, 2001.

[5] Bjørn Fjellvoll. Sensistivity analysis of the impact of geological uncertainties on production. http://www.nr.no/en/SAIGUP, 2003. Accessed: 2015-05-22.

[6] C. Klein and A. Philpotts. *Earth Materials: Introduction to Mineralogy and Petrology.* Earth Materials: Introduction to Mineralogy and Petrology. Cambridge University Press, 2012.

[7] Eduarda Lacerda, CM Da Silva, Viachteslav Ivanovich Priimenko, Adolfo Puime Pires, et al. Microbial eor: a quantitative prediction of recovery factor. In *SPE Improved Oil Recovery Symposium.* Society of Petroleum Engineers, 2012.

[8] L.W. Lake. *Enhanced oil recovery.* Prentice Hall, 1989.

[9] L.W. Lake. *Petroleum Engineering Handbook.* Society of Petroleum Engineers, 2007.

[10] I Lazar, IG Petrisor, and TF Yen. Microbial enhanced oil recovery (meor). *Petroleum Science and Technology*, 25(11):1353–1366, 2007.

[11] Jian Li, Jishan Liu, Michael G Trefry, Jungho Park, Keyu Liu, Bashirul Haq, Colin D Johnston, and Herbert Volk. Interactions of microbial-enhanced oil recovery processes. *Transport in porous media*, 87(1):77–104, 2011.

[12] K.-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST).* SINTEF ICT, 2014.

[13] Simon L Marshall. Fundamental aspects of microbial enhanced oil recovery: A literature survey. *National Research flagship, CSIRO, Western Australia*, 2008.

[14] Richard G. Miller and Steven R. Sorrell. The future of oil supply. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2006), 2013.

[15] Ann Muggeridge, Andrew Cockin, Kevin Webb, Harry Frampton, Ian Collins, Tim Moulds, and Peter Salino. Recovery rates, enhanced oil recovery and technological limits. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2006), 2013.

[16] Sidsel Marie Nielsen, Alexander Shapiro, Michael Locht Michelsen, and Erling Halfdan Stenby. 1d simulations for microbial enhanced oil recovery with metabolite partitioning. *Transport in Porous Media*, 85(3):785–802, 2010.

[17] Sidsel Marie Nielsen, Alexander Shapiro, Erling Halfdan Stenby, and Michael Locht Michelsen. *Microbial Enhanced Oil Recovery - Advanced Reservoir Simulation.* PhD thesis, Technical University of Denmark (DTU), 2010.

[18] Xavier Raynaud. Private communication May 2015.

[19] Statoil. Microbial enhanced oil recovery (meor). http://www.statoil.com/en/TechnologyInnovation/OptimizingReservoirRecovery/RecoveryMethods/ WaterAssistedMethodsImprovedOilRecoveryIOR/Pages/MicrobialEnhancedOilRecovery(MEOR).aspx, 2014. Accessed: 2015-04-15.

[20] Martin Thullner. Comparison of bioclogging effects in saturated porous media within one-and two-dimensional flow systems. *Ecological Engineering*, 36(2):176–196, 2010.

[21] MR Todd, WJ Longstaff, et al. The development, testing, and application of a numerical simulator for predicting miscible flood performance. *Journal of Petroleum Technology*, 24(07):874–882, 1972.

[22] Nathalie Tufenkji. Modeling microbial transport in porous media: Traditional approaches and recent developments. *Advances in Water Resources*, 30(6):1455–1469, 2007.

[23] B. Volesky. *Sorption and biosorption.* BV Sorbex, 2003.

[24] Stephen Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in Porous Media*, 1(1):3–25, 1986.