1
Introduction

Subsurface flow phenomena cover some of the most important technological challenges of our time. The road toward sustainable use and management of the earth's freshwater reserves necessarily involves modeling of hydrological systems in order to understand fluid movement in groundwater basins, quantify limits of sustainable use, monitor transport of pollutants in the subsurface, and appraise schemes for groundwater remediation. An equally important problem is the reduction of the emission of greenhouse gases like CO_2 into the atmosphere. Carbon sequestration in subsurface rock formations has been suggested as a possible means to that end. The primary concern is how fast the injected CO_2 will escape back to the atmosphere. Repositories do not necessarily need to store CO_2 forever, just long enough to allow the natural carbon cycle to reduce the atmospheric CO_2 to near pre-industrial levels. Nevertheless, making qualified estimates of leakage rates from potential storage facilities is a nontrivial task, and demands interdisciplinary research and software based on state-of-the-art numerical methods for modeling subsurface flow. Other obvious concerns are whether the injected CO_2 will leak into freshwater aquifers, or migrate to habitated or different legislative areas.

A third challenge is petroleum recovery. The civilized world will likely continue to depend on utilization of petroleum resources both as an energy carrier and as raw material for a wide variety of petrochemical products (fertilizers, detergents, dyes, synthetic rubber, plastics, etc.) in the foreseeable future. In the North Sea region, the recovery of existing and discovery of new conventional petroleum resources has declined significantly in recent years. Optimal utilization of known resources is therefore of utter importance to meet the demands for petroleum and lessen the pressure on exploration in vulnerable areas like in the arctic regions. Meanwhile, there has been a dramatic increase in the utilization of unconventional petroleum resources (shale gas/oil), and there is a strong need to understand how these can be produced in an economic way that minimizes harm to the environment.

Reliable computer modeling of subsurface flow is much needed to overcome these three challenges, but is also needed to exploit deep geothermal energy or to inject compressed gas

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

in the subsurface to store energy, ensure safe storage of nuclear waste, improve remediation technologies to remove contaminants from the subsurface, etc. Indeed, the need for tools that help us understand flow processes in the subsurface is probably greater than ever, and increasing. More than fifty years of prior research in this area has led to some degree of agreement in terms of how subsurface flow processes can be modeled adequately with numerical simulation technology.

This book introduces and discusses mathematical models describing flow processes in porous rocks on a macroscopic scale. The presentation focuses primarily on physical processes that take place during hydrocarbon recovery. Simulation of such processes is often referred to as reservoir simulation. This means that even though the mathematical models, numerical methods, and software implementations presented can be applied to any of the applications just mentioned, the specific examples use vocabulary, physical scales, and balances of driving forces specific to petroleum recovery. As an example of this vocabulary, we can consider the ability of a porous medium to transmit fluids. In petroleum engineering this is typically given in terms of the "permeability," which is a pure rock property, whereas one in water resource engineering is more concerned with the "hydraulic conductivity" that also takes the viscosity and density of the fluid into account. In CO_2 sequestration you can see both quantities used.

As an example of physical scales, let us compare oil recovery by water flooding and the long-term geological storage of CO_2. Petroleum resources can only accumulate when the natural upward movement of hydrocarbons relative to water is prevented by confinements in the overlying rocks. An oil reservoir is therefore usually a relatively closed system with a spatial extent of hundreds to thousands of meters, from which oil can be recovered over tens of years. The most promising candidates for geological CO_2 storage are huge aquifer systems that stretch out for hundreds of kilometers. The injection of CO_2 is mainly driven by pressure differences as in oil recovery. However, as CO_2 moves into the aquifer and the effects of the injection pressure ceases, the fluid movement is gradually dominated by buoyant forces that cause the lighter CO_2 phase to migrate upward in the open aquifer system. This process can potentially continue over thousands of years. The basic flow physics and governing equations are the same in both cases, but the balances between physical forces are different, and this should be accounted for when formulating the overall mathematical models and appropriate numerical methods.

Models and numerical methods developed to study subsurface flow are also applicable to other natural and man-made porous media such as soils, biological tissues and plants, electrochemical devices like batteries and fuel cells, concrete and other porous constructional materials, food and sanitary products, textiles, industrial filtering processes, polymer composites, water desalination, etc. Porous media models are also used to describe human physiology, e.g., waterways in the brain or flow in the capillary part of the vascular system. An interesting example is in-tissue drug delivery, where the challenge is to minimize the volume swept by the injected fluid, unlike in petroleum recovery, where one seeks to maximize the volumetric sweep to push out as much petroleum as possible.

1.1 Petroleum Recovery

We start by a conceptual sketch of the basic mechanisms by which hydrocarbon can be recovered from a subsurface reservoir. In good reservoir rock, the void spaces between mineral grains form networks of connected pores that can store and transmit large amounts of fluids. Petroleum discoveries vary in size from small pockets of hydrocarbon that may be buried just a few meters beneath the surface of the earth and can easily be produced, to huge reservoirs[1] stretching out several square kilometers beneath remote and stormy seas. However, as a simple mental picture, you can think of a hydrocarbon reservoir as a bent, rigid sponge that is confined inside an insulating material and has all its pores filled with hydrocarbons, which may appear in the form of a liquid oleic and a gaseous phase as illustrated in Figure 1.1. Natural gas will be dissolved in oil under high volumetric pressure like carbon dioxide inside a soda can. If the pressure inside the pristine reservoir is above the bubble point, the oil is undersaturated and still able to dissolve more gas. If the pressure is below the bubble point, the oil will be fully saturated with gas and any excess gas will form a gas cap on top of the oil since it is lighter.

To extract oil from the reservoir, we start by drilling a well into the oil zone. If the pristine pressure inside the reservoir is sufficiently high, it will push oil up to the surface and start what we will refer to as *primary production*. Alternatively, one may use a pump to lower the pressure in the wellbore below the point where oil starts flowing. How large

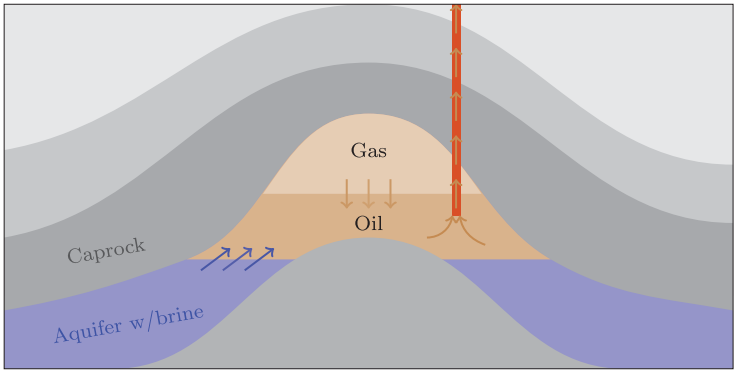

Figure 1.1 Conceptual illustration of a petroleum reservoir during primary production. Over millions of years, hydrocarbons have accumulated under a caprock that has a low ability to transmit fluids and thus prevents their upward movement. Inside the trap, the fluids will distribute according to density, with light hydrocarbons in a gaseous phase on top, heavier hydrocarbons in an oleic phase in the middle, and brine at the bottom. If the pressure difference between the oil zone and the wellbore is sufficiently high, the oleic phase will flow naturally out of the reservoir. As the oleic phase is produced, the pressure inside the reservoir will decline, which in turn may introduce other mechanisms that help to maintain pressure and push more oil out of the well.

[1] The largest reservoir in the world is found in Ghawar in the Saudi Arabian desert and is approximately 230 km long, 30 km wide, and 90 m thick.

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

pressure differential one needs for hydrocarbons to flow, depends on the permeability of the rock; the higher the permeability, the easier the hydrocarbons will flow toward the well.

As hydrocarbons are extracted, the pressure inside the reservoir will decay and the recovery will gradually decline. However, declining pressure will often induce other physical processes that contribute to maintain recovery:

- In a *water drive*, the pore space below the hydrocarbons is filled with salt water, which is slightly compressible and hence will expand a little as the reservoir pressure is lowered. If the total water volume is large compared with the oil zone, even a small expansion will create significant water volumes that will push oil toward the well and hence contribute to maintain pressure. Sometimes the underlying water is part of a large aquifer system that has a natural influx that replenishes the extracted oil by water and maintains pressure.
- *Solution gas drive* works like when you shake and open a soda can. Initially, the pristine oil will be in a pure liquid state and contain no free gas. Extraction of fluids will gradually lower the reservoir pressure below the bubble point, which causes free gas to develop and form expanding gas bubbles that force oil into the well. Inside the well, the gas bubbles rise with the oil and make the combined fluid lighter and hence easier to push upward to the surface. At a certain point, however, the bubbles may reach a critical volume fraction and start to flow as a single gas phase, which has lower viscosity than the oil and hence moves faster. This rapidly depletes the energy stored inside the reservoir and causes the production to falter. Gas coming out of solution can also migrate to the top of the structure and form a gas cap above the oil; the gas cap pushes down on the liquid oil and hence helps to maintain pressure.
- In a *gas cap drive*, the reservoir contains more gas than what can be dissolved in the oil. When pressure is lowered, the gas cap expands and pushes oil into the well. Over time, the gas cap will gradually infiltrate the oil and cause the well to produce increasing amounts of gas.
- If a reservoir is highly permeable, gravity will force oil to move downward relative to gas and upward relative to water. This is called *gravity drive*.
- A *combination drive* has water below the oil zone and a gas cap above that both will push oil to the well at the same time as reservoir pressure is reduced.

These natural (or primary) drive mechanisms can only maintain the pressure for a limited period and the production will gradually falter as the reservoir pressure declines. How fast the pressure declines and how much hydrocarbons one can extract before the production ceases, varies with the drive mechanism. Solution gas drives can have a relatively rapid decline, whereas water and gas cap drives are able to maintain production for longer periods. Normally only 30% of the oil can be extracted using primary drive mechanisms.

To keep up the production and increase the recovery factor, many reservoirs employ some kind of engineered drive mechanisms for *secondary production*. Figure 1.2 illustrates two examples of voidage replacement in which water and/or gas is injected to support pressure in the reservoir. Water can also be injected to sweep the reservoir, displace the oil, and push it toward the wells. In some cases, one may choose to inject produced formation

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

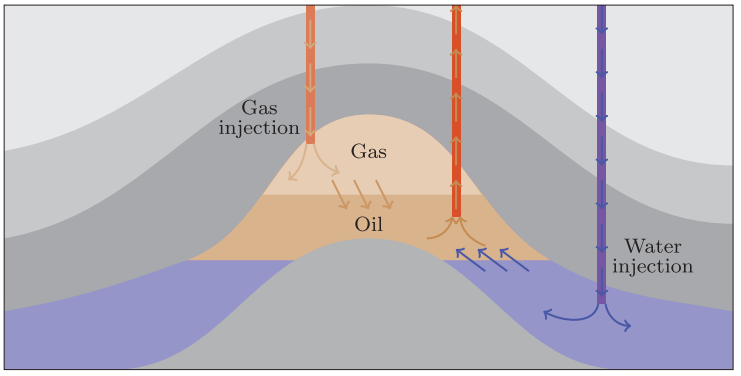

Figure 1.2 Conceptual illustration of voidage replacement, which is an example of a secondary production strategy in which gas and/or water is injected to maintain the reservoir pressure.

water, which is contaminated with hydrocarbons and solid particles and hence must be disposed of in some manner. Alternatively, one can extract formation water from a nearby aquifer. In offshore production it is also common to inject seawater. A common problem for all waterflooding methods is to maximize the sweep efficiency so that water does not move rapidly through high-flow zones in the reservoir and leave behind large volumes of unswept, mobile oil. Maintaining good sweep efficiency is particularly challenging for reservoirs with high-viscosity oil. If the injected water has lower viscosity than the resident oil, it will tend to form viscous fingers that rapidly expand through the oil and cause early water breakthrough in producers. (Think of water being poured into a cup of honey.) To improve the sweep efficiency, one can add polymers to the water to increase its viscosity and improve the mobility ratio between the injected and displaced fluids. Polymers have also been used to create flow diversions by plugging high-flow zones so that the injected fluid contacts and displaces more oil. For heavy oils, adverse mobility ratios can be improved by using steam injection or some other thermal method to heat the oil to reduce its viscosity.

Polymer injection and steam injection are examples of methods for so-called *enhanced oil recovery* (EOR), also called *tertiary production*. Another example is miscible and chemical injection, in which one injects solvents or surfactants that mix with the oleic phase in the reservoir to make it flow more readily. The solvent may be a gas such as carbon dioxide or nitrogen. However, the most common approach is to inject natural gas produced from the reservoir when there is no market that will accept the gas. Surfactants are similar to detergents used for laundry. Alkaline or caustic solutions, for instance, can react with organic acids occuring naturally in the reservoir to produce soap. The effect of all these substances is that they reduce the interfacial tension between water and oil, which enables small droplets of oil that were previously immobile to flow (more) freely. This is the same type of process that takes place when you use detergent to remove waxy and greasy stains from textiles. A limiting factor of these methods is that the chemicals are quickly adsorbed and lost into the reservoir rock.

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

Often, one will want to combine methods that improve the sweep efficiency of mobile oil with methods that mobilize immobile oil. Miscible gas injection, for instance, can be used after a waterflood to flush out residually trapped oil and establish new pathways to the production wells. *Water-alternating-gas* (WAG) is the most successful and widely used EOR method. Injecting large volumes of gas is expensive, and by injecting alternating slugs of water, one reduces the injected volume of gas required to maintain pressure. Similarly, the presence of mobile water reduces the tendency of the injected gas to finger through the less mobile oil. In polymer flooding, it is common to add surfactants to mobilize immobile oil by reducing or removing the interface tension between oil and water, and likewise, add alkaline solutions to reduce the adsorption of chemicals onto the rock faces.

Whereas the mechanisms of all the previously described methods for EOR are reasonably well studied and understood, there are other methods whose mechanisms are much debated. This includes injection of low-salinity water, which is not well understood, even though it has proved to be highly effective in certain cases. Another example is microbial EOR, which relies on microbes that digest long hydrocarbon molecules to form biosurfactants or emit carbon dioxide that will reduce interfacial tension and mobilize immobile oil. Microbial activity can either by achieved by injecting bacterial cultures mixed with a food source, or by injecting nutrients that will activate microbes that already reside in the reservoir.

Use of secondary recovery mechanisms has been highly successful. On the Norwegian continental shelf, for instance, water flooding and miscible gas injection have helped to increase the average recovery factor to 46%, which is significantly higher than the worldwide average of 22%. In other parts of the world, chemical methods have proved to be very efficient for onshore reservoirs having relatively short distances between wells. For offshore fields, however, the potential benefits of using chemical methods are much debated. First of all, it is not obvious that such methods will be effective for reservoirs characterized by large inter-well distances, as rapid adsorption onto the pore walls generally makes it difficult to transport the active ingredients long distances into a reservoir. Chemicals are also costly, need to be transported in large quantities, and consume platform space.

Even small improvements in recovery rates can lead to huge economic benefits for the owners of a petroleum asset and for this reason much research and engineering work is devoted to improving the understanding of mobilization and displacement mechanisms, and to design improved methods for primary and enhanced oil recovery. Mathematical modeling and numerical reservoir simulation play key roles in this endeavor.

1.2 Reservoir Simulation

Reservoir simulation is the means by which we use a numerical model of the geological and petrophysical characteristics of a subsurface reservoir, the (multiphase) fluid system, and the production equipment (wells and surface facilities) to analyze and predict how fluids flow through the reservoir rock to the stock tank or transport pipeline over time. It is

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

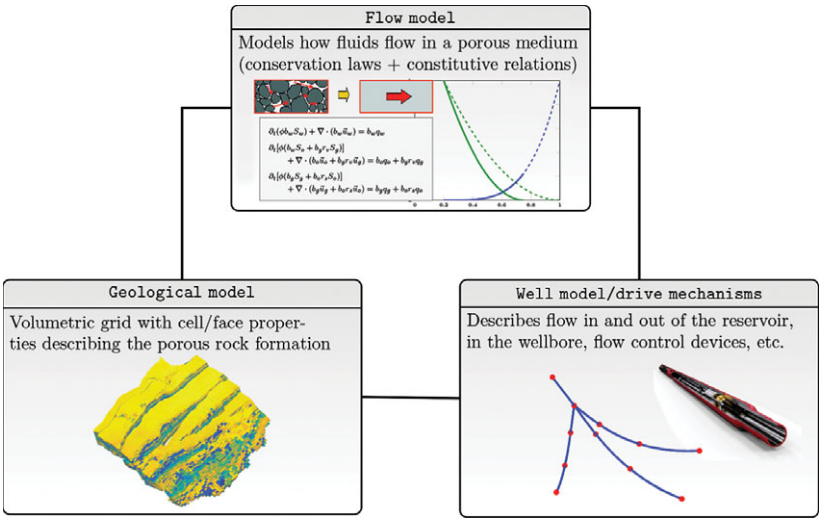

Figure 1.3 The three main constituents of a reservoir simulation model.

generally very difficult to observe and understand dynamic fluid behavior inside a reservoir, describe the physical processes, and measure all the parameters that influence the flow behavior. Predicting how a reservoir will produce over time and respond to different drive and displacement mechanisms therefore has a large degree of uncertainty attached. Simulation of petroleum reservoirs started in the mid 1950s as a means to quantify and reduce this uncertainty, and has become an important tool for qualitative and quantitative prediction of the flow of fluid phases. Reservoir simulation is a complement to field observations, pilot and laboratory tests, well testing, and analytical models, and is used by reservoir engineers to investigate displacement processes, compare and contrast the characteristics of different production scenarios, or as part of inverse modeling to calibrate reservoir parameters by integrating static and dynamic (production) data. In the big picture, reservoir simulation is mostly used to guide two different types of decisions: (i) to optimize development plans for new fields, and (ii) to assist with operational and investment decisions.

To describe the subsurface flow processes mathematically, our simulation model will be made up of three main constituents. First, we need a mathematical *flow model* that describes how fluids flow in a porous medium. These models are typically given as a set of partial differential equations describing the mass conservation of fluid phases, accompanied by a suitable set of constitutive relations that describe the relationship among different physical quantities. Second, we need a *geological model* that describes the given porous rock formation (the reservoir). The geological model is realized as a grid populated with petrophysical properties that are used as input to the flow model, and together they make up the reservoir simulation model. Last, but not least, we need a model for the *wells and production facilities* that provide pressure and fluid communication between the reservoir and the surface.

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

Accurate prediction of reservoir flow scenarios is a difficult task. One reason is that we can never get a complete and accurate characterization of the rock parameters that influence the flow pattern. Even if we did, we would not be able to run simulations that exploit all available information, since this would require a tremendous amount of computer resources that far exceed the capabilities of modern multiprocessor computers. On the other hand, we neither need nor do we seek a simultaneous description of the flow scenario on all scales down to the pore scale. For reservoir management it is usually sufficient to describe the general trends in the reservoir flow pattern.

In the early days of the computer, reservoir simulation models were built from two-dimensional slices with 10^2–10^3 Cartesian grid cells representing the whole reservoir. In contrast, contemporary reservoir characterization methods can model the porous rock formations by the means of grid-blocks down to the meter scale. This gives three-dimensional models consisting of millions of cells. Stratigraphic grid models, based on extrusion of 2D areal grids to form volumetric descriptions, have been popular for many years and are the current industry standard. However, more complex methods based on unstructured grids are gaining in popularity.

Despite an astonishing increase in computer power and intensive research on computation techniques, commercial reservoir simulators are rarely used to run simulations directly on geological grid models. Instead, coarse grid models with grid-blocks that are typically ten to hundred times larger are built using some kind of upscaling of the geophysical parameters. How one should perform this upscaling is not trivial. In fact, upscaling has been, and probably still is, one of the most active research areas in the oil industry. This effort reflects the general opinion that given the ever increasing size and complexity of the geological reservoir models, one cannot generally expect to run simulations directly on geological models in the foreseeable future.

Along with the development of better computers, new and more robust upscaling techniques, and more detailed reservoir and fluid characterization, there has also been an equally significant development in the area of numerical methods. State-of-the-art simulators employ numerical methods that can take advantage of multiple processors, distributed memory computer architectures, adaptive grid refinement strategies, and iterative techniques with linear complexity. For the simulation, there exists a wide variety of different numerical schemes that all have their pros and cons. With all these techniques available, we see a trend where methods from the research forefront are being tuned to a special set of applications and mathematical models, as opposed to traditional methods that were developed for a large class of differential equations.

Altogether, these developments have enabled geologists and reservoir engineers to build increasingly complex geological and reservoir models. Continuing to improve such models to gain a better understanding of the reservoir and reduce uncertainty is obviously important. Nevertheless, you should not forget that the purpose of a simulation study usually is to help your company make better decisions, and thus the value of the study generally depends on to what extent it influences decisions and leads to higher profit, e.g., by increasing recovery and/or reducing capital expenditures (CAPEX) or operational expenses (OPEX).

Developing a very advanced model that has all the physical effects imaginable included may therefore not be necessary to answer a specific question. Indeed, to paraphrase the famous Occam's principle: *a simulation model should be as simple as possible, but not simpler*. In the coming years, we will probably see a larger degree of hybrid modeling that combines models based on physical principles with data-driven approaches.

1.3 Outline of the Book

The book is intended to serve several purposes. First of all, you can use it as a self-contained introduction to the physics of flow in porous media, its basic mathematical theory, and the numerical methods used to solve the underlying differential equations. Hopefully, the book will also give you a hands-on introduction to practical modeling of flow in porous media, focusing in particular on models and problems that are relevant to the petroleum industry. The discussion of mathematical models and numerical methods is accompanied by a large number of illustrative examples, ranging from idealized and highly simplified setups to cases involving models of real-life reservoirs. Last, but not least, the book will introduce you to a widely used open-source software, and teach you some of the principles that have been used in developing it.

The Software Aspect: User Guide, Examples, and Exercises

All examples in the book have been created using the MATLAB Reservoir Simulation Toolbox (MRST). This open-source software consists on one hand of a set of reservoir simulators and workflow tools that you can modify to suit your own purposes, and on the other hand it can be seen as an enhancement of MATLAB/GNU Octave toward reservoir modeling in the form of a large a collection of flexible and efficient software libraries and data structures, which you can use to design your own simulators or computational workflows. The use of MRST permeates more traditional textbook material, and the book can therefore be seen as a user guide to MRST, in which you get introduced to the software gradually, example by example. Appendix A gives a more focused introduction to MRST, which starts by explaining how the software is organized, how you can install it, explore its functionality, and find help much in the same way as in MATLAB/GNU Octave.

The book can alternatively be viewed as a discussion by example of how a numerical scripting language like MATLAB/GNU Octave can for be used for rapid prototyping, testing, and verification on realistic reservoir problems with a high degree of complexity. Through the many examples, I also try to gradually teach you some of the techniques and programming concepts that have been used to create MRST, so that you later can use similar ideas to ensure flexibility and high efficiency in your own programs.

In the introductory part of the book that covers grids, petrophysical parameters and basic discretizations and solvers for single-phase flow, I have tried to make all code examples as self-contained as possible. To this end, all code lines necessary to produce the numerical results and figures are presented and discuss in detail. However, occasionally I omit minor details that either have been discussed elsewhere or should be part of your basic MATLAB

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

repertoire. As we move to more complex examples, in particular for multiphase flow and reservoir engineering workflows, it is no longer expedient to discuss scripts in full detail. In most cases, however, complete scripts containing all code lines necessary to run the examples can be found in a dedicated `book` module in MRST. I strongly encourage that you use your own computer to run as many as possible of the examples in the book, as well as other examples and tutorials that are distributed with the software. Your understanding will be further enhanced if you also modify the examples, e.g., by changing the input parameters, or extend them to solve problems that are related, but (slightly) different. To help you in this direction, I have included a number of computer exercises that modify and extend some of the examples, combine ideas from two or more examples, or investigate in more detail aspects that are not covered by any of the worked examples. For some of the exercises you can find solution proposals in the `book` module.

Finally, I point out that MRST is an open-source software, and if reading this book gives you ideas about new functionality, or you discover things that are not working as they should or could, you are free to improve the toolbox and expand it in new directions. If you do so, I strongly encourage you to pay us back by releasing your code publicly for the benefit of the reservoir simulation community.

Part I: Geological Models and Grids

The first part of the book discusses how to represent a geological medium as a discrete computer model that can be used to study the flow of one or more fluid phases. Chapter 2 gives you a crash course in petroleum geology and geological modeling, written by a non-geologist. In this chapter, I try to explain key processes that lead to the formation of a hydrocarbon reservoir, discuss modeling of permeable rocks across multiple spatial scales, and introduce you to the basic physical properties used to describe porous media in general. For many purposes, reservoir geology can be represented as a collection of maps and surfaces. However, if the geological model is to be used as input to a macroscale fluid simulation, we must assume a continuum hypothesis and represent the reservoir in terms of a volumetric grid, in which each cell is equipped with a set of petrophysical properties. On top of this grid, one can then impose mathematical models that describe the macroscopic continuum physics of one or more fluid phases flowing through the microscopic network of pores and throats between mineral grains that are present on the subscale inside the porous rock of each grid block.

Chapter 3 describes in more detail how to represent and generate grids, with special emphasis on the types of grids commonly used in reservoir simulation. The chapter presents a wide variety of examples to illustrate and explain different grid formats, from simple structured grids, via unstructured grids based on Delaunay tessellations and Voronoi diagrams, to stratigraphic grids represented on the industry-standard corner-point format or as 2.5D and 3D unstructured grids. The examples also demonstrate various methods for generating grids that represent plausible reservoir models, and they discuss some of the realistic data sets that can be downloaded along with the MRST software.

Through Chapters 2 and 3, you will be introduced to the data structures used to represent grids and petrophysical data in MRST. Understanding these basic data structures, and the various methods used to create and manipulate them, is fundamental if you want to understand the inner workings of a majority of the routines implemented in MRST or use the software as a platform to implement your own computational methods. Through the many examples, you will also be introduced to various functionality in MRST for plotting data associated with cells and faces (interface between two neighboring cells) as well as various strategies for traversing the grid and picking subsets of data; these techniques will prove very useful later in the book.

Part II: Single-Phase Flow

The second part of the book is devoted entirely to single-phase flow and will introduce you to many of the key concepts for modeling flow in porous media, including basic flow equations, closure relationships, auxiliary conditions and models, spatial and temporal discretizations, and linear and nonlinear solvers.

Chapter 4 starts by introducing the two fundamental principles necessary to describe flow in porous media: conservation of mass and Darcy's law. Mass conservation is a fundamental physical property, whereas Darcy's law is a phenomenological description of the conservation of momentum that introduces permeability, a rock property, to relate volumetric flow rate to pressure differentials. The chapter outlines different forms these two equations can take in various special cases. To form a full model, the basic flow equations must be extended with various constitutive laws and extra equations describing external forces that drive fluid flow; these can either be boundary conditions and/or wells that inject or produce fluids. The last section of Chapter 4 introduces the classical two-point finite-volume method, which is the current industry standard for discretizing flow equations. In particular, we demonstrate how to write this discretization in a very compact way by introducing discrete analogues of the divergence and gradient operators. These operators will be used extensively later when developing solvers for compressible single and multiphase flow with the AD-OO framework.

Chapter 5 focuses on the special case of an incompressible fluid flowing in a completely rigid medium, for which the flow model can be written as a Poisson-type *partial differential equation* (PDE) with a varying coefficient. We start by introducing the various data structures that are necessary to make a full simulator, including fluid properties, reservoir states describing the primary unknowns, fluid sources, boundary conditions, and models of injection and production wells. We then discuss in detail the implementation of a two-point pressure solver, as well as upwind solvers for linear advection equations, which, e.g., can be used to compute time lines (time-of-flight) in the reservoir and steady-state tracer distributions that delineate the reservoir into various influence regions. We end the chapter with four simulation examples: the first introduces the classical quarter five-spot pattern, which is a standard test case in reservoir simulation, while the next three explain how to incorporate boundary conditions and Peaceman well models and discuss the difference between using structured and unstructured grids.

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

Grids describing real reservoirs typically have unstructured topology and irregular cell geometries with high aspect ratios. Two-point discretizations are unfortunately only consistent if a certain relationship between the grid and the permeability tensor is satisfied, which is quite restrictive and difficult to fulfill when modeling complex geology. Chapter 6 discusses methods that are consistent for general polyhedral cell geometries. This includes mixed finite-element methods, multipoint flux approximation methods, and mimetic finite-difference methods that are still being researched by academia.

Chapter 7, the last in Part II, is devoted to compressible flow, which in the general case is modeled by a nonlinear, time-dependent, parabolic PDE. Using this relatively simple model, we introduce many of the concepts that will later be used to develop multiphase simulators of full industry-standard complexity. To discretize the transient flow equation, we combine the two-point method introduced for incompressible flow with an implicit temporal discretization. The standard approach to solving the nonlinear system of discrete equations arising from complex multiphase models is to compute the Jacobian matrix of first derivatives for the nonlinear system, and use Newton's method to successively find better approximations to the solution. Deriving and implementing analytic expressions for Jacobian matrices is both error-prone and time-consuming, in particular if the flow equations contain complex fluid models, well descriptions, thermodynamical behavior, etc. In MRST, we have therefore chosen to construct Jacobian matrices using automatic differentiation, which is a technique to numerically evaluate the derivatives of functions specified by a computer program to working precision accuracy. Combining this technique with discrete averaging and differential operators enables you to write very compact simulator codes in which flow models are implemented almost in the same form as they are written in the underlying mathematical model. This greatly simplifies the task of writing new simulators: all you have to do is to implement the new model equations in residual form and specify which variables should be considered as primary unknowns. When the software then evaluates all the elementary operations necessary to compute the residual, it simultaneously uses elementary differentiation rules to compute the analytical derivative of each elementary operation at the specific function value. These values are then gathered using the standard chain rule and assembled into a block matrix containing the correct partial derivatives with respect to all primary variables. To demonstrate the utility and power of the resulting framework, we discuss how one can quickly change functional dependencies in the single-phase pressure solver and extend it to include new functional dependencies like pressure-dependent viscosity, thermal effects, or non-Newtonian fluid rheology.

Part III: Multiphase Flow

The third part of the book outlines how to extend the ideas from Part II to multiphase flow. Chapter 8 starts by introducing new physical phenomena and properties that appear for multiphase flows, including fluid saturations, wettability and capillary pressure, relative permeability, etc. With this introduced, we move on to outline the general flow equations for multiphase flow, before we discuss various model reformulations and (semi-)analytical solutions for the special case of immiscible, two-phase flow. The main difference between

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

single and multiphase flow is that we now, in addition to an equation for fluid pressure or density, get additional equations for the transport of fluid phases and/or chemical species. These equations are generally parabolic, but will often simplify to or behave like hyperbolic equations. Chapter 9 therefore introduces basic concept from the theory of hyperbolic conservation laws and introduces a few classical schemes. We also introduce the basic discretization that will be used for transport equations later in the book.

Chapter 10 follows along the same lines as Chapter 5 and explains how incompressible solvers developed for single-phase flow can be extended to account for multiphase flow effects using the so-called fractional-flow formulation introduced in Chapter 8 and simulated using sequential methods in which pressure effects and transport of fluid saturations and/or component concentrations are computed in separate steps. The chapter includes a number of test cases highlighting various effect of multiphase flow and illustrating error mechanisms inherent in sequential simulation methods.

Once the basics of incompressible, multiphase flow has been discussed, we move on to discuss more advanced multiphase flow models, focusing primarily on the black-oil formulation, which can be found in contemporary commercial simulators. The black-oil equations lump all hydrocarbons into two pseudo-components: light components that exist in a gas phase at surface conditions and heavier components that exist in a liquid oil phase. At reservoir conditions, the two components form a gaseous and an oleic phase, but can also dissolve fully or partially in the other phase. The dissolution depends on pressure (and temperature), and the simplified pressure, volume, and temperature (PVT) behavior of the resulting fluids follows analytical or tabulated relationships. Together with water, the hydrocarbon system forms a three-phase, three-component system, whose behavior can be surprisingly intricate. Chapter 11 gives an in-depth discussion of the underlying physical principles, whereas Chapter 12 discusses how one can use variants of the automatic-differentiation methods introduced in Chapter 7, combined with modern principles of object orientation, to develop robust simulators. The two chapters also include a number of illustrative simulation cases.

Part IV: Reservoir Engineering Workflows

The fourth and last part of the book is devoted to discussing additional computational methods and tools that can be used to address common tasks within reservoir engineering workflows.

Chapter 13 is devoted to flow diagnostics, which are simple numerical experiments that can be used to probe a reservoir to understand flow paths and communication patterns. Herein, all types of flow diagnostics will be based on the computation of time-of-flight, which defines natural time lines in the porous medium, and steady-state distribution of numerical tracers, which delineate the reservoir into subregions that can be uniquely associated with distinct sources of the inflow/outflow. Both quantities will be computed using finite-volume methods introduced in Chapters 4 and 5, and you do not have to read the chapters in between to understand the essential ideas of Chapter 13. The concept of flow diagnostics also includes several measures of dynamic heterogeneity, which can be used

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

as simple proxies for more comprehensive multiphase simulations in various reservoir engineering workflows including placement of wells, rate optimization, etc.

As you will see in Chapter 2, porous rocks are heterogeneous at a large variety of length scales. There is therefore a general trend toward building complex, high-resolution models for geological characterization to represent small-scale geological structures. Likewise, large ensembles of equiprobable models are routinely generated to systematically quantify model uncertainty. In addition, many companies develop hierarchies of models that cover a wide range of physical scales to systematically propagate the effects of small-scale geological variations up to the reservoir scale. In either case, one quickly ends up with geological models that contain more details than what can or should be used in subsequent flow simulation studies. Hence, there is a strong need for mathematical and numerical techniques for formulating reduced models, or communicating effective parameters and properties between models of different spatial resolution. Such methods are discussed in Chapters 14 and 15. Chapter 14 introduces data structures and various methods for partitioning a fine-scale grid into a coarse-scale grid. This chapter hardly assumes any familiarity with flow solvers and can be read directly after Chapters 2 and 3. However, before you continue to read about upscaling in Chapter 15, which refers to the process in which petrophysical properties in all cells that make up a coarse block are averaged into a single effective value for each coarse block, I suggest that you read the chapters about incompressible, single-phase flow.

After this quick tour of the book, you are probably eager to start digging into the material. Before continuing to the next chapter, we present a simple example that will give you a first taste of simulating flow in porous media.

1.4 The First Encounter with MRST

The purpose of this first example is to show the ten code lines needed to set up and solve simple flow problem. We also show how to visualize the geological model and the computed flow solution. To this end, we consider a very simple problem: compute the pressure variation $p(z)$ inside a $[0,1] \times [0,1] \times [0,30]$ m^3 rock column for a single-phase fluid of constant density ρ assuming a datum pressure $p(z_0) = 100$ bar at the top of the column. The solution is trivial and follows by integration,

$$p(z) = p(z_0) + \int_{z_0}^{z} g\rho \, dz = p(z_0) + g\rho(z - z_0). \tag{1.1}$$

Here, g is the gravity constant along the z axis, which by convention in reservoir simulation is assumed to point downwards so that z increases toward greater depth.

We will now compute the same solution using a simple incompressible flow solver. To this end, we need two of the three constituents discussed in Section 1.2: (i) a flow model describing the fluid behavior and (ii) a model describing the reservoir rock. As explained already, the flow model consists of an equation describing conservation of mass and Darcy's law, which essentially represents conservation of momentum:

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

$$\nabla \cdot \vec{v} = 0, \qquad \vec{v} = -\frac{K}{\mu}\big[\nabla p - \rho g \nabla z\big], \tag{1.2}$$

Darcy's law relates the flow rate $\vec{v}$ to the gradient of the flow potential $p - \rho g z$. The constant of proportionality is given as the ratio between the macroscopic rock permeability K and the fluid viscosity μ. By eliminating $\vec{v}$, we can reduce (1.2) to an elliptic Poisson equation, whose solution is given by (1.1) for constant μ and K.

Most examples in MRST feature some geological model, which by convention is created first. The basic part of a geological model is a grid describing the geometry of the reservoir rock, here chosen as a regular $1 \times 1 \times 30$ Cartesian grid. The grid is generated in two steps using standard routines from MRST:

```
G = cartGrid([1, 1, 30], [1, 1, 30]*meter^3);
G = computeGeometry(G);
```

The first function constructs the grid topology and cell geometries. The second function computes derived quantities such as cell volumes and centroids, areas, and normals of all cell faces, which you need if you want to solve flow equations on the grid. More details about grids are given in Chapter 3. To plot the grid, we can use the command

```
plotGrid(G); view(3);
```

A geological model should also describe petrophysical and other geological properties. In our case, the only parameter is the permeability, which is set to 100 millidarcy (md or mD). MRST also requires porosity (i.e., the void fraction of the bulk volume) to be given for all models. Since it is not used here, we set it to a typical value of 0.2:

```
rock = makeRock(G, 0.1*darcy(), 0.2);
```

MRST works in SI units and we must therefore be careful to specify the correct units for all physical quantities. The result of this construction is a MATLAB structure that contains two vectors of 30 elements each, giving the permeability and porosity in each cell. Petrophysical properties are discussed in more detail in Chapter 2. The grid colored by permeability values is plotted as follows:

```
cla, plotCellData(G, rock.perm)
```

To start representing the flow part of the model, we first need a structure to hold the reservoir state, which in the basic form will consist of the pressure and the flux across cell faces. We can construct an empty container with all values set to zero as follows

```
sol = initResSol(G, 0.0);
```

Since we are solving (1.2) on a finite domain, we must also describe conditions on all boundaries. The default assumption is that there will be no flow over the reservoir boundary ($\vec{v} \cdot \vec{n} = 0$). In our case, we also need to prescribe $p = 100$ bar at the top of the column

```
bc = pside([], G, 'TOP', 100.*barsa());
```

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

To discretize Darcy's law in (1.2), we use a standard two-point flux-approximation scheme, which relates the flux between two neighboring cells i and j to their pressure difference, $v_{ij} = -T_{ij}(p_i - p_j)$. The constant of proportionality T_{ij}, is called the transmissibility and is computed as the harmonic average of two other constants $T_{i,j}$ and $T_{j,i}$ associated with cell i and j, respectively. Each of these quantities on depend on geometrical grid properties and the permeability tensor of a single cell and can be computed once and for all independent of the particular flow model once we have defined the grid and petrophysical parameters:

```
hT = computeTrans(G, rock);
```

This scheme will be discussed in more detail in Section 4.4. More advanced discretizations can be found in other add-on modules and are discussed further in Chapter 6.

The next thing we need to define is the fluid properties. Unlike grids, petrophysical data, and boundary conditions, data structures for representing fluid properties are not part of the basic functionality of MRST. The reason is that the way fluid properties are specified is tightly coupled with the mathematical and numerical formulation of the flow equations, and may differ a lot between different types of simulators. Here, we have assumed incompressible flow and can therefore use fluid models from the `incomp` add-on module,

```
mrstModule add incomp
gravity reset on
fluid = initSingleFluid('mu', 1*centi*poise, 'rho', 1014*kilogram/meter^3);
```

The second line ensures that gravity is set to its default value, whereas the third line defines the fluid to have a density of 1014 kg/m^3, which is representative for water. The constant dynamic viscosity does not affect the pressure distribution and is arbitrarily set to 1 cP. More details about the `fluid` object will be given in Section 5.1.1. All we need to know here is that we can query it for viscosity and density by the following call:

```
[mu,rho] = fluid.properties();
```

We now have all we need to set up and solve a discrete version of (1.2). The first sub-equation says that the fluxes across all faces of each cell should sum to zero, i.e., $\sum_j T_{ij}(p_i - p_j) = 0$, for all i. For Cartesian grids and homogeneous permeability, this scheme coincides with the classical seven-point finite-difference scheme for Poisson's problem and is the only discretization in the `incomp` module. As a final step, we use the transmissibilities, the fluid object, and the boundary conditions to assemble and solve the discrete system:

```
sol = incompTPFA(sol, G, hT, fluid,'bc', bc);
```

Having computed the solution, we plot the pressure given in units "bar," which equals 0.1 MPa and is referred to as "barsa" in MRST since "bar" is a built-in command in MATLAB:

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press

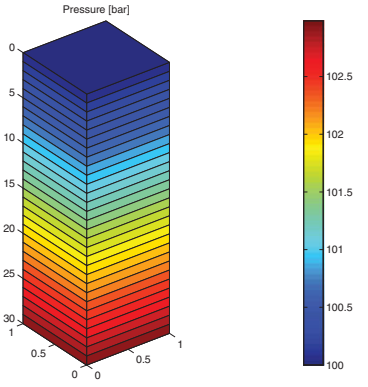

Figure 1.4 Hydrostatic pressure distribution in a gravity column computed by MRST. This example is taken from the MRST tutorial `flowSolverTutorial1.m` (`gravityColumn.m` in older versions of MRST).

```
plotFaces(G, 1:G.faces.num, convertTo(sol.facePressure, barsa()));
set(gca, 'ZDir', 'reverse'), title('Pressure [bar]')
view(3), colorbar, set(gca,'DataAspect',[1 1 10])
```

From the plot shown in Figure 1.4, we see that our solution correctly reproduces the linear pressure increase given in (1.1).

Before you continue reading, I encourage you to consult Appendix A, which describes MRST in more detail and explains how to install and get started with the software. Having the software operational is not a prerequisite for reading this book, but will contribute significantly to increase your understanding. And in case you should worry, you do not need to type in all the individual MATLAB/MRST commands; complete source codes for almost all the cases discussed later in the book can be found in a special add-on module, which you activate by calling `mrstModule add book`.

https://doi.org/10.1017/9781108591416.002 Published online by Cambridge University Press