Preface

The book you are holding is a follow-up to *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave* published by Cambridge University Press in 2019, which gives a unique introduction on reservoir simulation, with its strong focus on computational aspects and how you would implement all the methods in practice. It also serves as a user guide to the MATLAB Reservoir Simulation Toolbox (MRST), and we thus herein refer to it as the "MRST textbook." In the current book, we expand on the material from the MRST textbook by explaining more recent features in MRST for rapid prototyping and improved computational performance and teaching the mathematical models and computational methods underlying new modules aiming at more accurate and efficient simulations or more advanced reservoir physics.

MRST: From Internal Research Tool to a Community Code

The MATLAB Reservoir Simulation Toolbox (MRST) has come a long way since the Computational Geosciences group at SINTEF first started developing it in late 2007. Originally, the software was intended as an internal toolbox to support the development of consistent discretizations and multiscale mixed finite-element methods on stratigraphic and unstructured grids. To be able to test new methods on the type of grids used by industry, we had to spend quite some time developing robust grid-processing and visualization routines for corner point, perpendicular bisector (PEBI), and other types of unstructured grids. We also developed a variety of data structures to represent petrophysical parameters, fluid behavior, boundary conditions, wells, etc. SINTEF has a long tradition of developing software tools of professional quality as part of our contract research, and it was therefore natural for us to put some effort into code quality and documentation even for an internal prototyping tool. The toolbox quickly proved to be quite versatile in terms of grid complexity, but we doubted that it had much commercial value, because it lacked many of the essential features seen in contemporary simulators.

https://doi.org/10.1017/9781009019781.001 Published online by Cambridge University Press

However, we nonetheless thought that it might have some value for students and other researchers, and in early 2009 we decided to give our toolbox a name and release it as free, open source, using a license that would prevent others from turning (parts of) it into a commercial product. (To be honest, we also harbored a small hope that free access to the new mimetic and multiscale methods would contribute to trigger additional interest from industry.)

So, on April 1, 2009, MATLAB Reservoir Simulation Toolbox version 1.1 was published online. The first release consisted of 147 MATLAB scripts, containing 8 594 lines of comments and 6 747 lines of code according to `cloc`. The toolbox offered three pressure solvers (two-point, mimetic, and multiscale mixed finite elements) and two transport solvers (implicit/explicit), both for incompressible flow, as well as a set of user tutorials and routines for 2D/3D plotting, grid processing, setting up data structure, unit conversion, etc.

Since then, the toolbox has gradually developed into a community code with a worldwide user base, as evidenced by the more than 190 master/PhD theses and 430 external scientific papers that we know for certain have used it.[1] Starting in 2011, MRST has followed a biannual release cycle, and since 2018 the underlying software repositories have been publicly available on Bitbucket.

Over the 23 published releases, the software has grown quite a lot, and the latest version, MRST 2021a, consists of a core module with basic tools for rapid prototyping of new computational methods plus 56 different add-on modules, large and small, that address a wide variety simulation and modeling needs. Sixteen of these modules have either been fully or partially developed by people outside of SINTEF. Altogether, the code base counts 3 387 MATLAB files with 246 000 lines of code and more than 117 000 lines of comments. Adding to this, there are now 3.3 times as many C/C++ code lines for MEX-accelerated functionality as the number of MATLAB code lines in the original release. You can also find third-party modules online that for various reasons are not included in the official release (e.g., because they have not been kept fully up to date or because the authors want to distribute the code themselves). There are also several modules in development, externally or internally in SINTEF, that we hope will be added to future releases.

Altogether, MRST has become a powerful prototyping tool for developing, validating, and verifying new computational methods and modeling methodologies. It is therefore natural that many modules implement methodologies that are still at the research front and have not yet been adopted in engineering practice. On the

[1] You can find an updated list of theses and publications on the MRST website. The list is far from complete and consists of work that Google Scholar or the authors have notified us of or work we have stumbled upon while reading scientific literature. We thus expect the real number to be higher, in particular for master/PhD theses.

other hand, the software also offers a lot of functionality found in commercial engineering tools, including industry-grade simulation capabilities for black-oil and compositional models. As a result, we see that an increasing number of users wish to apply MRST as if it was a standard simulator. Significant effort has therefore been invested into making MRST more computationally efficient and scalable, by removing computational overhead and integrating more high-performing linear solvers; this is discussed in more detail in Chapter 6.

One feature that distinguishes MRST from many other commercial tools is that any simulator built using the object-oriented, automatic differentiation (AD-OO) framework is differentiable or can quite easily be configured to be so, which means that users can obtain sensitivities and gradients with respect to model parameters. This is particularly useful for applications in optimization, model calibration, and uncertainty quantification. In fact, MRST supported solution of adjoint equations even before the AD-OO framework was introduced, and many of our primary users therefore came from research groups working on various aspects of long-term production optimization.

Back in 2009, the original scope of MRST was to develop new simulation technology for Darcy-type, single-continuum flow processes on a reservoir scale; e.g., as encountered in hydrocarbon recovery from clastic rocks. Since then, the scope has been significantly broadened. One of the first new applications introduced was modeling of long-term CO_2 storage in large-scale saline aquifers. To derive detailed trapping inventories over thousands of years, we developed a comprehensive set of modeling capabilities including geometric methods for deriving static capacity estimates, spill-point methods for simple dynamic estimates, as well as vertically integrated flow models, adaptively coupled with local 3D models, for accurate simulation of all trapping mechanisms except for mineral trapping. Likewise, the first third-party contribution included in the official release of MRST came in 2012 in the form of a module for discrete fracture modeling (DFM). Later, modeling of fractured reservoirs became much more developed and now includes modules for hierarchical and embedded discrete fracture modeling, dual-porosity modeling, multi-continuum models, and modeling of unconventional shale oil/gas. Likewise, new add-on modules have also been introduced for simulating geochemistry, geomechanical, and geothermal processes.

An important part of making a community code is to ensure that the software is properly documented, not only in terms of tutorials and inline or help-line documentation of functions and key data structures but also in terms of technical user guides that describe the underlying methods and models and discuss potential applications and important limitations. Early on, one of us (K-A) started writing such a user guide. Finishing it took almost 10 years, primarily because MRST kept expanding and hence also the scope of the book. At some point we had to draw the line and what we herein will refer to as the MRST textbook was published in 2019 by Cambridge University Press under Gold Open Access thanks to a generous donation from Equinor. The book gives a thorough introduction to the models, methods, and design principles underlying key parts of MRST, including grids and petrophysics, incompressible flow simulation (the `incomp` family of modules), compressible (black-oil) models, rapid prototyping and the AD-OO framework, as well as three selected workflow elements (grid coarsening, upscaling, and flow diagnostics to better understand the dynamic heterogeneity and volumetric communication in a reservoir). Although the book is quite long (650+ pages), it leaves large parts of the software undocumented. Even before the book was officially published, it was clear to us that we at some point would have to write a follow-up to satisfy the many requests for user guides for the more advanced parts of MRST. We quickly decided that such a book would have to be a multiauthored volume focusing on individual modules of the software.

The Need for This Book

Before the summer of 2019, we therefore put out a public call for contributed chapters to a new book on advanced functionality developed using MRST. The call stated that each chapter should be written in textbook form and should function as a user guide and tutorial for a specific module (what constitutes a module in MRST is explained in the next paragraph). To this end, the chapter should motivate the module; explain why it is interesting, and what types of problems it can be applied for; teach the methods and/or models implemented; and go through a few selected examples that outline the main functionality. The style should be much like in the first MRST book; e.g., with code excerpts intermingled with background theory and examples the readers can run themselves. All results should be reproducible, so that each chapter should be accompanied by a full set of code and data that are part of the official MRST releases and should be publicly accessible on Bitbucket.

To further explain the background of the book, a module in MRST is, strictly speaking, a collection of functions, object declarations, and example scripts located in a folder. Each module needs to have a unique name and reside in a different folder than other modules. Our only requirements are that the code is well tested and documented in a format that does not deviate too far from that used elsewhere in MRST; uses a clear naming convention that avoids potential clashes with other parts of MRST; and contains a few tutorial examples that outline the main functionality and explain the most common syntax. The code also needs to contain a clear specification of copyright and the license under which it can be used (the GNU General Public License). In addition, we recommend that modules do not use functionality from MATLAB's many toolboxes, which potential users may not have access to.

https://doi.org/10.1017/9781009019781.001 Published online by Cambridge University Press

Following these specifications, we received close to 30 chapter proposals, and 17 of these were later developed into full manuscripts. After a careful peer and editorial review of the manuscripts and the accompanying code, 14 chapters were approved for inclusion in the book. These chapters span a wide variety of applications and research directions and have been subdivided into three parts.

Overview of the Book

Part I of the book focuses on grid generation, discretizations, and solvers and consists of four chapters. Chapter 1 discusses generation of constrained Voronoi grids that adapt to line and surface constraints. Using the methods in this chapter, you can easily generate complex unstructured grids that adapt to fault surfaces, fractures, or well paths. Chapters 2 and 3 discuss new methods for spatial discretization: nonlinear finite-volume schemes to ensure consistent pressure discretization that preserve the monotonicity of the solution, and discontinuous Galerkin discretization for improved spatial resolution of transport terms, which, e.g., can be used to reduce numerical smearing of linear and weakly nonlinear waves. Chapter 4 explains in detail the type of multiscale finite-volume methods that recently have been implemented in the commercial INTERSECT simulator and shows how this technology can be used to accelerate reservoir simulation.

Part II discusses recent developments in MRST that have been introduced to improve prototyping capabilities and ensure that the computational cost of running large simulation cases is more comparable with that of simulators written in compiled languages. Chapter 5 explains the concept of state functions and outlines further steps that have been taken to modularize the AD-OO framework. As a result, any simulator written using this framework can now be viewed as a differentiable graph in which the individual components, like fluid and pressure–volume–temperature properties or spatial/temporal discretizations, can easily be extended or replaced; e.g., to change the spatial and temporal discretization. State functions also play an essential part in the implementation of simulators for chemical enhanced oil recovery (EOR) and compositional simulations in Chapters 7 and 8. Chapter 6 discusses new backends that reduce the computational overhead of automatic differentiation as well as use of external, high-end iterative solvers that aim to improve MRST's scalability to larger models. The chapter also outlines tools for setting up and managing simulations cases so that aborted simulations can be restarted and results from previous simulations can be quickly retrieved without having to repeat the full simulation.

Part III concerns modeling of new physical processes beyond the simple incompressible and compressible black-oil models discussed in the MRST textbook. Chapter 7 gives a quick introduction to the salient physical mechanisms for surfactant and polymer flooding, which are two primary examples of chemical EOR

https://doi.org/10.1017/9781009019781.001 Published online by Cambridge University Press

techniques. The implementation in MRST relies heavily on the new state-function concept, and Chapter 5 is therefore strongly suggested as a pre-read. The chapter also reviews basic fractional-flow analysis and explains why trailing chemical waves are particularly susceptible to numerical smearing. Chapter 8 outlines the compositional module of MRST, whose development initially motivated the introduction of state functions and new accelerated AD backends.

Chapters 9–11 all concern modeling of fractured media. By reading Chapter 9, you will learn about embedded discrete fracture models (EDFMs), in which the fracture network is represented implicitly on a lower-dimensional grid constructed independently of the grid that represents the solid rock (matrix). Applicability of EDFMs is demonstrated on three examples, including a stochastically generated fracture network and a data set sourced from the Jandaira carbonate formation in Brazil. Chapter 10 introduces you to two other types of fracture modeling approaches. In discrete fracture and matrix (DFM) modeling, fractures are represented explicitly as lower-dimensional elements in the grid, whereas multi-continuum models represent the fractures and matrix as distinct continua that interact through transfer functions. The chapter presents a unified modeling framework that enables you to develop hybridized models that combine both approaches. The framework makes no distinction between dual-continuum and DFM methods and treats fractures and matrix as flowing or virtual domains, with transfer functions reinterpreted as fluxes between cells in different domains.

In Chapter 11, the authors combine and extend functionality from the modules for compositional flow and EDFM to model storage and transport mechanisms in fractured unconventional oil and gas reservoirs. Such reservoirs consist of organic-rich source rocks and have very low matrix permeability and porosity. Hydrocarbons are usually produced by inducing hydraulic fractures, propped by solid particles, that connect the inherent natural fracture networks to the horizontal production wells to extract hydrocarbons stored in void spaces in the rock matrix, in microcracks, and in the natural fractures. The chapter discusses how to model the specific storage and transport mechanisms of unconventional reservoirs, including storage of gas in the sorbed state, contributions from molecular diffusion to the mass flux, and permeability changes in the fractured rock induced when pressure drops during production.

In enhanced geothermal systems (EGS), fractures are induced in a region with low permeability and high temperature and serve the same purpose as the fins of a conventional heat exchanger. Chapter 12 presents a new MRST module for modeling low- to moderate-enthalpy geothermal systems such as EGS and high-temperature aquifer thermal energy storage (HT-ATES).

Chapters 13 and 14 introduce you to some of the physics governing deformable rocks. Chapter 13 discusses unsaturated flow in non-deformable and deformable

https://doi.org/10.1017/9781009019781.001 Published online by Cambridge University Press

porous media, modeled by the Richards' equation and the equations of unsaturated poroelasticity, respectively. Accurate modeling of such flow processes has high relevance in environmental sciences, hydrogeology, soil mechanics, and agriculture, and in the chapter, you will be introduced to the pertinent mathematical models and a new family of multipoint finite-volume solvers. Chapter 14 concerns the combined effects of flow and geomechanics, as modeled by a full linear poroelastic system. Inclusion of geomechanical effects is important to model processes such as fracturing pressure, fault (re)activation, seismicity, and subsidence. The chapter teaches the basic principles of geomechanics and its coupling to flow and outlines various solution strategies, including fully coupled, sequentially split, and fixed-stress split schemes.