Reference this work:

Al Ibrahim, M. A., 2019, Petroleum system modeling of heterogeneous organic-rich mudrocks, PhD Thesis, Stanford University, p. 131-135.

THE AUTOMATED BASIN AND PETROLEUM MODELING TOOLBOX FOR PROBLISTIC INTERACTIONS STUDIES

Mustafa A. Al Ibrahim, Stanford University

Abstract

We present am open source library for automating Schlumberger PetroMod software using MathWorks MATLAB. The library is designed to be user friendly by abstracting most work from the casual user. It allows the user to duplicate and modify lithologies, duplicate and modify models, simulate models, and retrieve the simulation outputs. With the automated functionality, it is now possible to run large number of variable models for new type of studies such as ones involving uncertainty analysis, sensitivity analysis and machine learning related studies.

Keywords: PetroMod, MATLAB toolbox, automation.

Introduction

Basin and petroleum system modeling is a powerful tool to study the evolution of basins and the formation of source rocks. Because of the complexity involved in simulating basins, researchers, and geoscientists in general, rely on already developed commercial software. One of the most commonly used is Schlumberger PetroMod. PetroMod, while powerful, is not script friendly. This makes its usage for multiple research and practical scenarios that requires running large number of models difficult, while not impossible. The patch functionalities in the software is limited to certain parameters and does not allow great flexibility. This functionality and flexibility are especially important nowadays where uncertainty analysis, sensitivity analysis and machine learning techniques require running larger numbers of simulations with variable inputs.

To this end, we present a Matlab toolbox that can be used to automate most of PetroMod functionalities. Example applications of the code is it use in running large number of simulations in 1) research studies

such as heterogeneity and upscaling studies, 2) practical studies such as running multiple scenarios to optimize calibration fit. This paper will present a brief description of the library capabilities, suggested workflows for usage, and a description of the code and its organization to facilitate code extensions and modification. Examples of the library usage are included with source code and will not be covered here for brevity.

Library features

The library is not designed as a substitute for the graphical user interface in PetroMod. Instead it is designed to complement it. The user is expected to create the initial template model in the graphical interface as usual. The library was heavily tested using one- two, and three-dimensional models using PetroMod 2017.2 or later. Current available functionalities are:

- Duplicate and delete lithologies.
- Modify any parameter in the lithologies including curves.
- Mix lithologies with default mixers used in PetroMod. Users can also design their own custom mixers.
- Duplicate and delete models.
- Modify model parameters including boundary curves and horizons.
- Modify simulation parameters.
- Run models.
- Read present day output files.
- Run Open Simulator scripts (given that the user has a license for Open Simulator).

Suggested workflow

The workflow suggested for using the library is staggered over a number of steps. These are: 1) define the parameters for all the models simulated, 2) update the lithology file with all the new lithologies that will be used, 3) create all the models but duplicating a template model and modifying it by the parameters defined in step 1, 4) simulate the models, and 5) read the outputs of the models. Note that the staggered workflow will allow for manual checks between steps to make sure that the code is working as expected.

There are situations where this staggered workflow is not possible for resources limitations, for example, when models are large and cannot all be stored at the same time, or for technical limitations, for example,

when running a Markov Chain Monte Carlo steps where each simulation would rely on the results of the previous step. In this case, it is required to update each model, simulate them, and obtain the results sequentially.

The Workshop folder accompanying the source code provides a guided tour of the library and the suggested workflow. General knowledge of the Matlab language and language concepts, such as classes, is required. Some parameters are stored with units that are different than what they appear in graphic user interface and so care must be taken when modifying the parameters. In addition, the library is not designed to detect all erroneous inputs and so it is the responsibility of the user to make sure that inputs are correct, for example, the library will not check if the input is negative for a porosity value.

Code organization

This section is written for users who want to extend the code, but it should be valuable for the casual user too in getting familiar with the library. The code interacts with PetroMod in two ways: 1) by modifying a PetroMod project files directly, and 2) by calling some of PetroMod executables, specifically the simulator and the Open Simulator script runner. The code is organized into three major components. The user is expected to usually interact with two the first two.

PetroMod class

This is the main class to control the general functionalities of PetroMod, such as lithology manipulation, running simulations, running scripts, and model management. Lithology manipulation is done by using functionalities in the Lithology class. Running simulations and scripts is done by calling the PetroMod executables with their corresponding arguments, Hermes and runpmpy for the simulator and script executable respectively. Model management such as duplication and deletion is done by folder operations such as copying and deletion.

Model classes

The three main model classes that the user interact with are: Model1D, Model2D, and Model3D. They all inherit most of their methods from their parent class: Model. Model class contains functionalities such as loading model parameters from model files as variables, updating the parameters, and saving parameters back to their files. Model1D, Model2D, and Model3D defines dimension specific files that need to be

loaded. The file types used are PetroMod specific formats and are handled by specific classes (see description below).

Lithology file class

Lithology is stored as an Extensible Markup Language file (XML) in the definition folder of the project (named def). Note that PetroMod does not create the file unless a modification is made and saved in the lithology editor. The XML lithology file is composed of three main nodes: 1) Meta, 2) Curve, and 3) Lithology. The Meta node stores information about the parameters including unique ids that are used to identify them elsewhere in the files, names, units, etc. The Curve node stores all the curves for all the lithology, e.g., porosity-permeability relationship. Lithology stores the lithology information. Note that Lithology and Curve nodes do not store parameters names and so the Meta node can act as a look-up table.

The code for the Lithology file is stored in four classes: one class for each node with the same name and one class (LithoFile) to control them and relate them to each other. The user controls the LithoFile through the PetroMod class. Finally, a class called LithoMixer is used to create lithology mixtures. Note that custom LithoMixer classes can be created and used. Note that Lithology classes are the oldest part of the code and could benefit from code refactoring for readability and ease of extension.

Support classes

Several static support classes exist as utilities for the functionalities of the toolbox: 1) FileTools for general file operations, 2) HashTools to create unique ids, 3) XMLTools to manage the lithology XML file, 4) StatsTools for simple statistical functions, and 5) a number of PetroMod specific file formats classes to support reading, modifying, and writing the files (PMATools, PMTTools, PMDTools, and PMDGroupTools). PMA is the PetroMod ASCII file and it contains text data. Although it is currently used with specific format, the files can generally be variable and so a modification of the class might be needed if other files need to be handled. PMT is PetroMod table file and it contains tabled information. PMD is the PetroMod decimal file and contains a vector of numbers. PMD Groups are a collection of PMD files that are indexed by using a PetroMod table file (PMT).

Final remarks

Automation in basin and petroleum system modeling opens new opportunities in research and simplifies current workflow in the oil and gas industry. Possible improvements to the toolbox include: 1) automatic

model dimension detection, 2) lithology code refactoring, and 3) thorough testing for working with three-dimensional models.

Acknowledgements

We would like to thank Anshuman Pradhan for his valuable assistance in testing the software and providing guidance on useful functionalities that should be added. We would also like to thank Tanvi Chheda for the suggestions made.

Software availability

The BPSM PetroMod Toolbox is available from the Stanford Rock Physics and Borehole Geophysics Project GitHub page under the BSD 3-Clause license (https://github.com/MosGeo/AutoBPSMToolbox).