PREDICT: PeRmEability DIstributions of Clay-smeared faulTs

MATLAB® r2020b | License GPLv3 | last commit today | DOI

Overview • Citation • Requirements • Download • Installation • Examples • License • Acknowledgements

---

This code was developed as part of my PhD in Prof. Ruben Juanes' lab (juanesgroup.mit.edu) at MIT Civil and Environmental Engineering.

NOTE: Several code files are not documented yet. I will be adding documentation in the file headers in September and October, 2022, and create a new release after the process is completed.

NOTE 2: Contributions are welcome, please send me a message first (lsalo@mit.edu).

Short Description

Computes upscaled fault permeability distributions using a parameter-based, probabilistic description of clay and sand smears. MATLAB and MRST are required to run this code.

Overview

PREDICT is a novel algorithm designed to *predict*, i.e., compute, the diagonal components of the fault permeability tensor (perpendicular to the fault, k_{xx}; strike-parallel, k_{yy}; and dip-parallel, k_{zz}) in relatively shallow siliciclastic sequences (<~3 km of depth). The computation is done for a given throw interval, and the material distributions and output permeability values are representative of the main shear zone within a faulted sediment volume. Hence, **PREDICT** represents an architectural domain typically referred to as the fault core (see figures below). PREDICT can be run in two modes: 2D and 3D.

2D workflow

1. The algorithm takes a set of numerical quantities, the input parameters, that describe the faulted stratigraphy.
2. **PREDICT** generates marginal probability distributions, according to the input parameters, for another set of numerical quantities; these are intermediate random variables required to compute the dimensions, distribution and permeability of the clay- and sand-smears. Samples are drawn accounting for variable dependency.
3. Object-based simulation is used to place the clay-smears within the fault (a fault cross section with x-z dimensions). The sand smears are placed next, according to the amount and location of the space available.
4. The equivalent or upscaled fault permeability computation for the modeled throw interval is based on the material distribution within the fault, and uses flow-based upscaling for k_{xx} and k_{zz}, while area-

weighted arithmetic average is used for k_{yy}.

Steps two to four are repeated multiple times, each one representing one realization or simulation, until the full permeability distribution for each component is obtained (see figure below). As shown below, the 2D version is significantly faster and less memory-intensive, but leads to lower permeabilities and cannot accurately model k_{yy}.
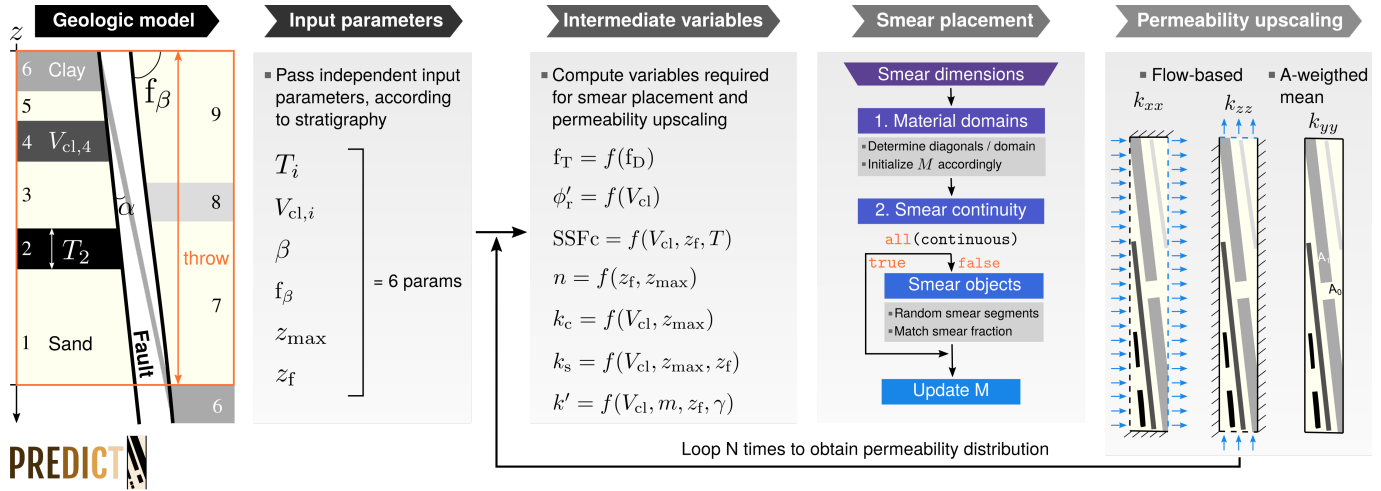

3D workflow

Steps one to three are identical to the 2D version (described above).

4. Multiple fault cross sections are constructed in the same way (i.e., following steps two-three). The number of cross sections is equal to s_s, and all of them are assigned the same fault thickness. This represents the inner loop in the workflow figure below.

5. A 3D fault volume is assembled by concatenating n=s_s sections from point 4 above. Flow-based upscaling is used for all three fault permeability components. Additionally, in the 3D version the user can specify a coarse grid for upscaling with multiple cells in either of the three dimensions, which is useful for subsequent reservoir simulation in faulted settings.

Similar to the 2D version, steps two to five are repeated multiple times, each iteration representing one realization or simulation, until the full permeability distribution for each component is obtained.
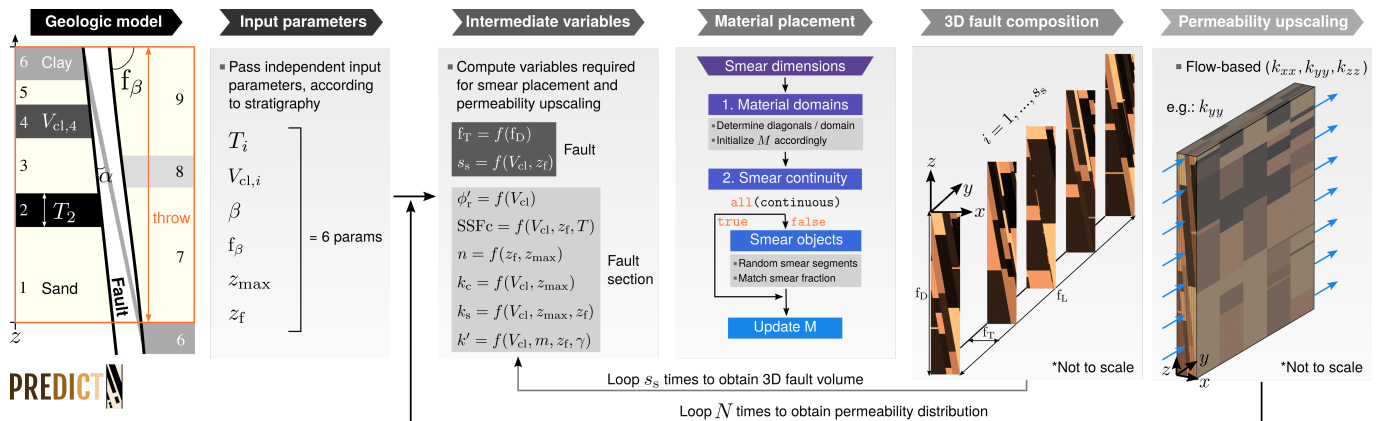
Citation

The full description of PREDICT is provided in the paper (and, in particular, the supplement) below. **To be updated**

Paper

```
@article{salo22faultpermPaper,
   title={Fault permeability from stochastic modeling of clay smears},
   author={Llu{\'i}s Sal{\'o}-Salgado and Davis, J Steven and Juanes,
Ruben},
   year={Accepted for publication in {\em Geology}}
}
```
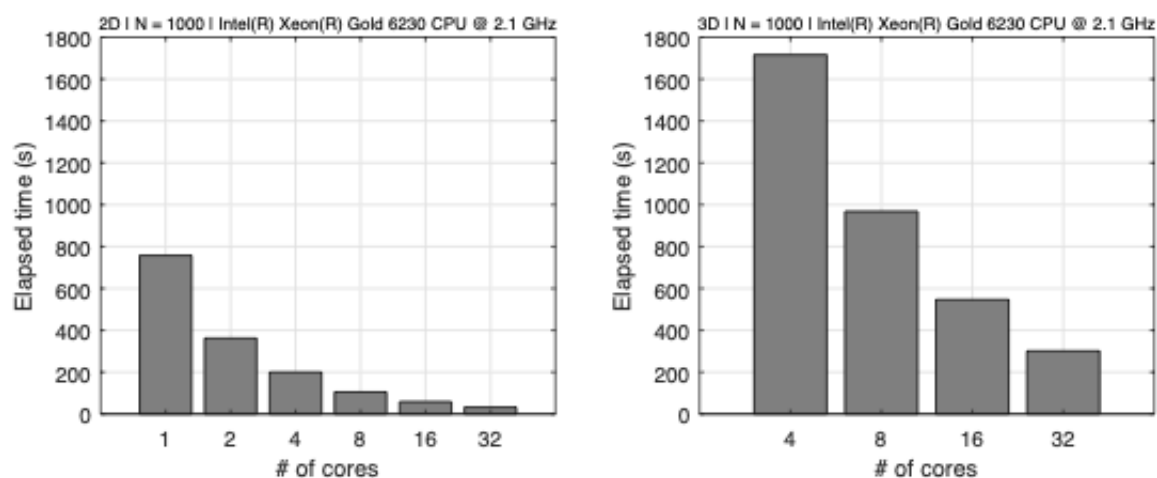
Code

```
@software{salo22faultpermCode,
   author = {Llu{\'i}s Sal{\'o}-Salgado and Davis, J Steven and Juanes,
Ruben},
   doi = {https://doi.org/10.5281/zenodo.7104886},
   title = {{PREDICT}},
   url = {https://github.com/lsalo/predict},
   version = {1.0.0},
   month = {09},
   year = {2022}
}
```

If PREDICT is useful for your research, please consider citing it.

Requirements

Hardware: When running thousands of simulations, a machine with multiple cores (4+) and few GB of RAM (6-8+) is recommended (2D version). For the 3D version, we recommend a machine with a minimum of 16GB RAM, given that some of the structures will be heavy, but it is possible to run the code with a 8GB RAM laptop. Running times are provided below for 1000 realizations in a given stratigraphy:

Software: PREDICT can be run on Windows, macOS and Linux (i.e. any OS where MATLAB can be installed). The code incorporates MRST functionality for flow-based permeability upscaling, so it requires an installation of both MATLAB and MRST (very straightforward, see steps below). Development took place using MATLAB r2020b and the development version of MRST. The code has also been tested with MATLAB

r2020a and r2021a, as well as the current MRST public release ([v2021a](#)). Backward compatibility with previous MATLAB versions also supported by MRST 2021a is likely, but it has not been tested.

In addition, PREDICT uses the following MATLAB add-on toolboxes:

- [Statistics and Machine Learning Toolbox](#) (**required**): For generating intermediate variable distributions and samples.
- [Parallel Computing Toolbox](#) (*recommended*): Not required, but *highly* recommended for anyone using the code beyond exploration purposes. Time gains when running parallel simulations are illustrated above for 1000 realizations/simulations of a given stratigraphic case.
- [Curve Fitting Toolbox](#) (*recommended*): Not required, but may be useful for output analysis.

Download

- **MATLAB**: Can be installed following the instructions [on the website](#), and your academic institution likely provides campus-wide access free of charge.
- **MRST**: The latest public release of MRST can be downloaded [here](#):
- **PREDICT**: The repository can be cloned or downloaded from [here](#) (green button "clone or download"). [**TBD**]

Installation

We show installation steps for both MRST and PREDICT.

1. Download the latest MRST release (see above).
2. Download PREDICT (see above).
3. From within MATLAB, run the `startup.m` file in the main MRST folder. You will be prompted whether you want to enable MEX acceleration, select enable. This can be changed later by running the command `mrstSettings()`.
4. From within MATLAB, right click on the `predict` folder and select "Add to Path > Selected Folders and Subfolders".

You can now run PREDICT (see Examples section below). Note that, for flow-based permeability upscaling, both a TPFA and a MPFA can be used in the 2D version. If you are running PREDICT without MEX, TPFA is recommended (otherwise, it will be slow).

Examples

Examples are provided in the folder [examples](#). For a comprehensive introductory example, run `example0_singleStrati.m` or, in the live editor, `example0_singleStrati_live.mlx` (2D). Similar files are provided for the 3D version:

Example 0: Single stratigraphic case + analysis (3D)

This is a complete introductory example. It shows how to load the appropriate MRST modules, define the inputs according to a given faulted stratigraphy, and generate the output permeability distributions. A comprehensive analysis of the results is also shown. The algorithm is run in 3D mode.

We first make sure that the workspace is clean:

```
clear
close all force
%rng('default')          % check repeatability
```

1. Load Required MRST Modules

First, navigate to the mrst folder and run `startup.m`. We can then load the appropriate modules for generating MRST grids and upscale the permeability:

```
mrstModule add mrst-gui coarsegrid upscaling incomp mpfa mimetic
mrstVerbose on      % set to on for more insight in the command window
```

2. Define Model and Upscale Permeability

```
% 2.1 Mandatory input parameters
% Footwall first and hangingwall next, e.g. {[footwall, FW], [hangingwall, HW]}.
% We need to define the layer thickness, clay content, layer dip angle, fault
% dip angle, faulting depth, and burial depth. Further details about input parameter
% formatting, etc can always be checked from the documentation in the classes
% and functions.
thickness = {repelem(25, 1, 4), [5 10 15 10 20 10 10 5 15]};              % [m]
vcl       = {[0.1 0.4 0.2 0.5], ...
             [0.3 0.6 0.1 0.7 0.2 0.8 0.3 0.9 0.1]};                      % fraction [-]
dip       = [0, 0];                                               % [deg.]
faultDip  = 70;                                                   % [deg.]
zf        = [500, 500];                                           % [FW, HW], [m]
zmax      = {repelem(2000, numel(vcl{1})), repelem(2000, numel(vcl{2}))};   % {FW, HW}
dim       = 3;                    % dimensions (2 = 2D, 3 = 3D)
unit_plot = 'm';

% 2.2 Optional input parameters
% In this case, we indicate a maximum fault material permeability of and a correlation
% coefficient for dependent variables:
maxPerm = 1000;               % [mD]
rho     = 0.6;                % Corr. coeff. for multivariate distributions

% 2.3 Flow upscaling options and number of simulations
U.useAcceleration = 1;            % 1 requires MEX setup, 0 otherwise (slower for MPFA).
U.method          = 'tpfa';       % 'tpfa' recommended for 3D
U.coarseDims      = [1 1 1];      % desired n cells [x, y, z] in coarse grid
U.flexible        = true;         % default true, much faster but U.coarseDims
                                  % will be modified in some realizations
                                  % unless U.coarseDims = [1 1 1] (do not set
                                  % to false in that case).
Nsim              = 1000;         % Number of 3D simulations/realizations
```

...

```
% 3.4. Visualize upscaled permeability
% Plot upscaled permeability distributions (all simulations)
plotUpscaledPerm(faults, dim)
```
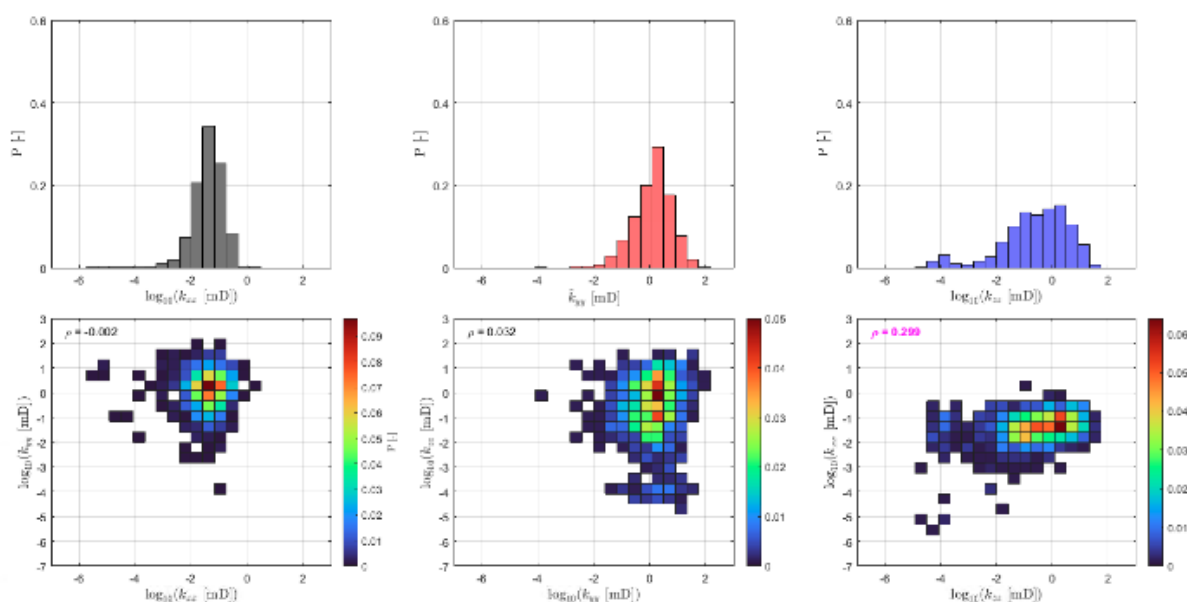

License

PREDICT incorporates MRST functionality, so it legally becomes an extension of MRST. Therefore, it is also released under the terms of the GPL license.

Acknowledgements

Readme design based on examples from Awesome README.