Comments on the DFM module for MRST

Introduction

This document contains various comments on the Discrete Fracture Matrix module, both in terms of implementation details and guidelines for the discretization and simulation of fracture media. The module is developed at the Department of Mathematics, University of Bergen mainly by Tor Harald Sandve. Some utility files are provided by Uni Research. For a short overview of the module confer the README file and examples in the DFM module.

Overview

The module provides control volume discretizations (two- and multi-point flux approximations) of fractured media by representing fractures as lower-dimensional objects that still are represented as cells in the computational grid. The discretization is described in (Karimi-Fard, et al., 2004; Sandve, et al., 2012). The approach assumes that the fractures are represented as faces in the grid, thus the module also provide some utility functions for triangulating fractured domains. Finally, the transport solvers (both explicit and implicit) in the MRST core are modified to account for fracture transport. MRST files that are modified have a suffix _DFM to distinguish them from the original files.

Known bugs / shortcomings

- Discretizations and solvers are modified from core MRST functions to account for fracture cells; look for files with suffix _DFM. The modifications are based on MRST version 2011a, and are thus static with respect to modifications of the core functions.
- A method for gridding general 3D fractures would have been nice. An import filter for TetGen might be a way to go.
- The MPFA flux calculation does not handle pyramids in 3D (more generally, each cell should share exactly Nd faces with each of its vertexes). This puts some constraints on the gridding tools in 3D.
- The code has mostly been tested for horizontal problems, and seems to work well there. When gravity is present, the TPFA routine seems to work there as well, whereas the MPFA method for this problem is barely tested, and should be used with caution. In particular, the elimination of small cells in fracture intersections is questionable.
- Capillary pressure has not been tested.
- The system for tags should be improved. See below.

Lessons learned in gridding of fractures

When discretizing a fractured media, ensuring the quality of the grid is highly important for the overall simulation result. The grids for the DFM model should be conforming, meaning that the fractures are represented as edges in the grid. This constrained gridding is non-trivial, see (Holm, et al., 2006) for examples of typical challenges. Below we give a summary of lessons learned while working with this, with the hope that it can be useful for others. Note that the observations should be considered tricks / approximations that in many cases can be justified, but nevertheless should be applied with caution:

- It is extremely useful to apply a somewhat relaxed notion of fracture positions, so that fractures that span multiple faces need not be represented as a straight line in the grid. This flexibility allows for much better cells close to the fracture, resulting in increased computational accuracy. This can be done by defining a resolution for the accuracy of the fracture representation, and snap all grid vertexes to this underlying grid. To get reasonably shaped cells, this resolution should not be too fine compared to the grid itself. Note that fractures that are close, but not connected may be connected in the computational grid due to the snapping. This can in some cases lead to critical differences in the flow properties of a fracture network. If so, the resolution for fracture representation, and possibly also the grid size in that area should be increased.
- An alternative to relaxing the fracture location is either to add vertices adaptively, as is done by the Triangle software (Shewchuk, 1996; Shewchuk, 2002). In our experience the resulting grids are not ideal for solving flow problems, although it works reasonably well. To be specific, the difference between TPFA and MPFA may be significant for these grids, even in isotropic media.
- In our experience, tags are an invaluable tool for discretizing and simulating on fractured media. With the current implementation each face can have a tag, which is stored in G.faces.tags (named so to avoid interfering with G.faces.tag which may be used for other purposes, more on this later). It is assumed throughout the module that all fracture faces have tag > 0, see for instance the transmissibility computations. Furthermore, hybrid cells will have a tag (G.cells.tags) which contains the index of the face it was created from. In practice it can be useful to apply multiple face tags with different meanings. If for instance we consider families of fractures, which all have multiple members, and it is of interest to identify which fracture a face belongs to, the face should have two tags: One for family number, and one for index within the family. This can be achieved by a complex numbering system, but a better approach is to let G.faces.tags be a matrix with arbitrary number of columns, and further introduce G.faces.tagcols as a cell of string identifiers for the columns in G.faces.tags. The same can be used for G.cells.

References

Holm, R. et al., 2006. Meshing of domains with complex internal geometries. *Num. Lin. Alg. Appl.,* 13(9), pp. 717-731.

Karimi-Fard, M., Durlofsky, L. & Aziz, K., 2004. An Efficient Discrete-Fracture Model Applicable for General-Purpose Reservoir Simulators. *SPE Journal,* pp. 227-236.

Sandve, T., I., B. & J.M., N., 2012. An efficient Multi-Point Flux Approximation based approarch for Discrete Fracture Matrix Simulations. *J. Comput. Phys.,* 231(9), pp. 3784-3800.

Shewchuk, J., 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: M. Lin & D. Manocha, eds. *Applied Computational Geometry: Towards Geometric Engineering.* s.l.:Springer Verlag, pp. 203-222.

Shewchuk, J., 2002. Delaunay Refinement Algorithms for Triangular Mesh Generation. *Computational Geometry: Theory and Applications,* 22(1-3), pp. 21-74.