Computers & Geosciences 102 (2017) 1–11

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

ADFNE: Open source software for discrete fracture network engineering, two and three dimensional applications

Younes Fadakar Alghalandis

Alghalandis Computing Co., New Westminster, BC, Canada

ARTICLE INFO

Keywords:
Fracture networks
DFN
Stochastic modeling
Open source
Matlab
Fluid flow
Simulations

ABSTRACT

Rapidly growing topic, the discrete fracture network engineering (DFNE), has already attracted many talents from diverse disciplines in academia and industry around the world to challenge difficult problems related to mining, geothermal, civil, oil and gas, water and many other projects. Although, there are few commercial software capable of providing some useful functionalities fundamental for DFNE, their costs, closed code (black box) distributions and hence limited programmability and tractability encouraged us to respond to this rising demand with a new solution. This paper introduces an open source comprehensive software package for stochastic modeling of fracture networks in two- and three-dimension in discrete formulation. Functionalities included are geometric modeling (e.g., complex polygonal fracture faces, and utilizing directional statistics), simulations, characterizations (e.g., intersection, clustering and connectivity analyses) and applications (e.g., fluid flow). The package is completely written in Matlab scripting language. Significant efforts have been made to bring maximum flexibility to the functions in order to solve problems in both two- and three-dimensions in an easy and united way that is suitable for beginners, advanced and experienced users.

1. Introduction

Fractures are everywhere. They occur in bones, natural or artificial materials, and literally in the entire nature; the most dominant exposures however are associated with rocks. Under variant critical internal and external stress conditions (i.e., stress regime) rock failures take place which in turn result in fractured domains (CFCFF, 1996). A fractured rock generally therefore refers to a domain that consists of intact parts of rock also called "rock blocks" and to separations between the blocks also called "fractures" (Goodman and Shi, 1985; Jing, 2003). A broader classification for fractures would include all types of separations in the rock such as faults, joints, bedding and so on.

Fractures are important as they play critical role in material strength, rock block stability, as well as in creating pathways for fluid and gas flow (Dverstorp, 1991; CFCFF, 1996; Berkowitz, 2002, Koyama et al., 2009, Fadakar-A et al. 2013b-2013c). In mining (Elmo et al., 2013, 2014), civil (Staub et al., 2002) and geothermal projects (Hanano, 2004; Wyborn et al., 2005; Grasby et al., 2012) it is extremely vital to study fractures for optimum and safe mineral exploitation, for designing proper support systems in tunnels (Hernqvist, 2009) and other underground works, and for modeling of fluid (heat) flow in heat chambers, respectively. It is also of great importance in oil and gas industry (Cosgrove, 1998; Nelson, 2001) particularly in unconventional reservoirs (shale gas) where through fractures (preexist or stimulated) the oil and gas is translated and extracted. In water reservoirs (Zimmerman and Bodvarsson, 1996; Singhal and Gupta, 2010) the extent and the quality of an aquifer is directly affected by the characteristics of fractures in the host and surrounding rocks. For mineral concentrations (Nelson, 2001) the presence of fractures prior to or during mineralization stage dictates the type (geneses, formation) and extent of reserves e.g., gold vein formation. Another important application is in nuclear waste disposal sites where the host rock is thoroughly investigated and continuously monitored for fractures and their connectivity to maintain safety requirements and to avoid catastrophic failures (Follin et al., 2006).

Fracture and fracture network modeling and simulations are active researches most notably in the last decade. The recent exponential growth in the modeling of fractured rock is greatly supported by rapid developments in the computing hardware and software. A typical model of small fracture network consists of thousands of fractures which expectedly results in several folds more complexity in inter-connectivity between fractures and other characteristics. The modeling and simulations of fractures and fracture network are commonly carried out in two- or three-dimensional space (Dershowitz et al., 2000; Jing and Stephansson, 2007; Fadakar-A, 2014). The choice of dimension is determined primarily by the nature of study and expected goals as well. Nevertheless, due to extensive complexity in geometrical modeling in three-dimension, researches have noticeably been tailored

E-mail address: younes.fadakar@yahoo.com.

http://dx.doi.org/10.1016/j.cageo.2017.02.002
Received 27 September 2016; Received in revised form 1 February 2017; Accepted 2 February 2017
Available online 04 February 2017

towards two-dimensional case studies (Huseby et al., 1997; Staub et al., 2002; Vogel, 2002; Koike and Ichikawa, 2006; Blocher et al., 2010). Some related recent works on three-dimensional fracture network modeling to name are Gringarten (1996); Merrien-Soukatchoff et al. (2012); Koike et al. (2015). It is worth noting here that there are few proprietary and commercial software applications that are capable of dealing with both two- and three-dimensional cases. Beside their high costs, the two most important considerations and limits in their use are, however, "closed source" releases and missing development capability for "end users". Where the later issue basically limits their potential use as reliable research tools, the former issue gives no chance for any development; hence, these limits significantly discourage conducting fundamental researches due to non-tractable results. To address these issues and also to help to popularize fracture network engineering concepts specifically in discrete formulation (Dershowitz et al., 2000; Fadakar-A, 2014), we here introduce Alghalandis Discrete Fracture Network Engineering (ADFNE), a comprehensive fracture and fracture network modeling software package which is open source Matlab readable code. It consists of 295 (and growing) functions, for general and specialized purposes, that work together seamlessly to handle variety of needs including geometrical modeling, model simulations, model characterizations and applications, and data exchange (importing and exporting). The package is aimed to elegantly and efficiently deal with both two and three dimensional use cases. All hard works remain in the background (source code is provided) in a way a novice user would interact quickly with the functions, while experienced users would explore the source code for further learning of the tactics and concepts implemented, any improvement and further development as it happened to become a need during research.

2. Discrete fracture network engineering

Discrete fracture network engineering (DFNE) deeply rooted in stochastic modeling (Kendall, 2003; Chiles, 2004; Fadakar-A, 2014) provides useful tools to characterize fractured rock for wide range of interests in research and industry including stability analysis of rock blocks (in rock mechanics and geotechnics, for example) and fluid flow modeling (in geothermal, oil and gas, groundwater, for example). Mathematically and statistically robust, the stochastic principles of DFNE and its comprehensive, flexible and scalable framework and tools ensure obtaining utmost information from even limited, sparse and often multi-type dataset (Fadakar-A et al., 2013a) which often is the case in fractured rock problems. Surface observations (scanline, compass measurements), subsurface loggings (Ozkaya and Mattner, 2003 from boreholes, tunnels etc.) or deep seismic event records (Tang et al., 1996; Fadakar-A et al., 2013a), whatever the data type is, the model can benefit from them during the establishment, calibration, validation and improvement of governing functions in every stage of simulation. DFNE tools can also be adapted such to utmost represent any need while preserving legitimacy, reliability and performance of every stage. This goal can be achieved due to modular structure of DFNE frameworks.

Based on the principles, every fracture is built discretely following some key rules such as: a fracture is a flat object, its shape, if not an infinite plane, is a convex polygon (rectangle, ellipse or more complex form), and its size follows a known distribution function such as negative exponential (Diggle, 2003; Baddeley, 2010). Similarly, its location is obtained by means of spatial functions such as two- or three-dimensional uniform or Poisson distributions. Spatial inhomogeneity (Chiles, 2004; Illian et al., 2008) can be applied to the fracture locations to impose nonstationary density of points. The orientation information can be extracted from uniform or Fisher distributions, for example. Further adjustments such as obtaining desired intersection system (e.g., fracture termination forms ×: two crossing fractures, ⋌: one terminates as it reaches another fracture, or ∧: both fractures terminate at the intersection) can also be utilized at this stage. Advanced refinements such as spatial clustering and connectivity can also be implemented by means of optimization tools such as simulated annealing (Andersson and Dverstorp, 1987; Deutsch and Cockerham, 1994).

In summary, fractures appear in many real world applications and hence advanced fracture network modeling is of great interest in research and industry communities. Recent developments in computing systems are pushing forward even further the growth of DFNE applications. Future perspective spread over all disciplines that are associated with fractured domain phenomena in any scale.

3. Alghalandis discrete fracture network engineering package

Despite the increasing interests in the application of DFNE concepts, available software tools are very limited, mainly due to the complicated code implementation of the concepts. Typically, a comprehensive tool for DFNE includes routines for building, inspection and processing of the geometry of fractures, topology and spatial distributions of fractures in two- and three-dimensional spaces. Any of these features is a challenging topic on its own. Considering them all together suggests big challenges, reportedly. Furthermore, due to the nature of problems in DFNE it is quite ordinary to work on fracture networks with populations over several thousand fractures. Efficient codes are therefore required. For couple of million or more in size, beside the optimized codes, parallel computing, cluster computing and access to super computers are to be considered.

1. Closed Code Solutions

   In the industry, few commercial packages are available for DFNE which are completely closed source and quite expensive as well. They also are limited by design to specific tasks with near to no functionality for broader researches. Some of them support internal scripting which basically works only for automatization of built-in processes. Developing of an idea foreign to the existing functionality of them is however impossible due to being closed source. Practically speaking, for implementation of emerging ideas the use of closed codes is quiet impractical if not impossible.
2. An Open Source Solution

ADFNE package is written in *Matlab* (Mathworks, 2016), a well-documented standard code scripting language (popular in both academia and industry), and is fully readable even by novice users. The open source package provides users unique opportunity to monitor what is going on in the code here or there, that is, there is no "black box" situation. Such a transparency in ADFNE (and under its easy and free of charge licensing, Fadakar-A, 2016) gives researchers even a chance to adapt the code into their particular interests whenever needed. As a result, further developments become plausible and fast as well. Furthermore, the philosophy behind the package structure design supports elegantly and efficiently the scalability for complex projects and delivers high performance such that many functions can be adapted with minimum efforts for parallel computing, hence, broader applications are even foreseeable. Worth noting that with a reasonable effort every function in ADFNE can be translated effectively into other popular programming language such as *Python* (Python Foundation, 2016). Indeed, we aim to implement full functionality of ADFNE in Python (as an open source programming language) in future works. It is also worth noting here the *Octave* open source project that can run *Matlab* scripts quite well.

3.1. Structure design

The current release of ADFNE package includes 295 functions from which 63 functions are specialized to deal with two-dimensional applications (Table 1) and 100 functions for three-dimension