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
0098-3004/

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 $\times$: two crossing fractures, $\curlywedge$: one terminates as it reaches another fracture, or $\wedge$: 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

Table 1
Functions for two-dimensional cases, few from 63 functions available in ADFNE.

Function	Description
BBoxLines2D.m	Bounding box of lines
BreakLinesX2D.m	Splits lines into segments at their intersections
CFi2D.m	*Connectivity Field* (Pixel Edition) for 2D fracture networks
ClipLinesByPoly2D.m	Clips 2D lines by a polygonal region
ConnectivityField2D.m	*Connectivity Field* for 2D fracture networks
ConnectivityIndex2D.m	Connectivity Index for 2D fracture networks
Density2D.m	*True Density* of 2D fracture networks
DrawPoly2D.m	Draws 2D polygons
ExtendCollapseFractures2D.m	Manipulate stochastically the length of 2D fractures
GeneralisedConnectivityField2D.m	Generalized Connectivity Field for 2D fracture networks
GenFNM2D.m	Simulates 2D fracture networks
GridXLines2D.m	Returns intersections between a grid and 2D lines
IsolatedLines2D.m	Isolation test for 2D lines
Lengths2D.m	Returns length of 2D lines
LineSimilarity2D.m	Evaluates similarity between 2D line sets
LinesToClusters2D.m	Clustering (grouping) information of 2D lines
LinesX2D.m	All intersections between 2D lines
LinesXLines2D.m	All intersections between two sets of 2D lines
RandLinesInPoly2D.m	Random 2D lines inside a 2D polygon
Resize2D.m	Resizes a 2D matrix (e.g., image)
SaveLinesAsHTML2D.m	Saves 2D fracture network in HTML format
SaveLinesAsSVG2D.m	Saves 2D fracture network in SVG format
SupCSup2D.m	Examines if two 2D support cells are connected
SupXLines2D.m	Intersections between a 2D support cell and lines

Table 2
Functions for three-dimensional cases, few from 100 functions available in ADFNE.

Function	Description
BBox3D.m	Bounding box of 3D points
Centroids3D.m	Centroids of 3D polygons (fractures)
ClassifyPipes3D.m	Classifies pipes into inlet, outlet and inner pipes
ClipPolys3D.m	Clips 3D polygons by a given domain (e.g., cube)
CompareDFNs3D.m	Compares 3D fracture networks for similarities
DrawGraph3D.m	Visualization of graph (nodes, edges)
DrawPipes3D.m	Visualization of pipe model
DrawPolys3D.m	Vsualization of fracture polygons
DrawSlices3D.m	Slice graph for volumetric data (such as CF)
Exchange3D.m	Exchanges fractures in the 3D fracture network (perturbation)
FNMPipes3D.m	Generates pipe model from 3D fracture network
GenFNM3D.m	Simulation of 3D fracture networks
IsolatedLines3D.m	Examines isolation of 3D fractures in the network
Lengths3D.m	Lengths (e.g., maximum size) of 3D fractures
Orientation3D.m	Orientation information of 3D fractures
PolyInfo3D.m	All geometrical and spatial information of 3D fractures
PolysX3D.m	All intersections between 3D fractures
PolysXPolys3D.m	Intersections between two sets of 3D fractures (polygons)
PolyToDipDir3D.m	Sets a 3D polygon to desired direction
RandPoly3D.m	Random polygon in space
Resize3D.m	Resizes volumetric data
SavePolysToVTK3D.m	Saves fracture network in VTK format
SaveToFile3D.m	Saves points as file
ScaleFNM3D.m	Scales fracture network
SetAxes3D.m	Sets graph axes into three-dimension
Size3D.m	Determines lengths of fractures
SupCSup3D.m	Evaluates connection between two supports
VolRender3D.m	Volumetric rendering e.g., for CF, GCF, etc.

Table 3
Generic functions, few from 132 functions available in ADFNE.

Function	Description
Backbone.m	Backbone (skeleton) of 2D/3D fracture networks
Bbox.m	Bounding box of input (points, lines etc.)
BoxPlotE.m	Boxplot graph
Clusters.m	Clustering (grouping) based on intersection information
CompareHistogram.m	Compares two histograms
ConnectivityMatrix.m	Connectivity Matrix
CosineSimilarity.m	Cosine similarity index for 2D lines
FNMToGraph.m	Graph structure from given fracture network
Histo.m	Draws histogram
Labels.m	Labels fractures based on clusters
P10fromP32.m	Generates P10 from P32
P21fromP32.m	Generates P21 from P32
P21G.m	Generates P21 grid edition
P21Pix.m	Generates P21 pixel edition
P22G.m	Generates P22 grid edition
RandFisher.m	Fisher random numbers
Relabel.m	Relabels cluster labels by their size
Rose.m	Rose diagram
SaveFig.m	Saves figure in high resolution
Scale.m	Scales data to desired bounds
Statistics.m	Returns statistics of the input
Stereonet.m	Draws Stereonet graph

applications (Table 2). The package also provides 132 generic functions (Table 3) that handle both dimensions delivering variety of functionalities including data import and export, visualizations (Goodman and O'Rourke, 2004; Vince, 2005; Freeden et al., 2010), data manipulation and analysis and special purposes.

The naming protocol for functions used in ADFNE package follows meaningful and descriptive, short, distinguishable, and extendable schemes. Briefly, if there was no dimension reference in the name (i.e., 2D or 3D) then the function is empowered to automatically handle both two- and three-dimensional data. In addition, some functions for three-dimensional cases have been developed over their two-dimensional function's structure, that is, similar stages are implemented to bring more readability and user friendliness to the entire package. This helps users to learn quickly and to have a good coverage of functionality in both dimensional applications.

3.2. Fracture network models

The simulation of fracture networks, characterization and applications all are straightforward by using ADFNE package. For example, a two-dimensional fracture network is easily simulated by means of GenFNM2D function as follows.

$$lines = GenFNM2D(n, theta, kappa, minl, maxl, rgn);$$

where n is the number of fractures, *theta* is the mean orientation $[0..2\pi)$, *kappa* is dispersion factor for the orientation (≥ 0), *minl* is minimum length (>0) and *maxl* is maximum length ($>minl$) for fractures, and *rgn* is the region of study $[x_{min}, x_{max}, y_{min}, y_{max}]$ by which the simulated fracture network is clipped. Note that, von Mises-Fisher distribution is used for orientation in which a value close to zero ($\ll 1$) for *kappa* results in omnidirectional orientation (almost Uniform distribution, U) for a fracture network. *Kappa* values higher than 5, observably, dictate the direction defined by *theta* for the entire network. The use of *minl* and *maxl* helps to avoid generating very short and very long fractures. The length value for each fracture is obtained from negative Exponential (E) distribution. Other distribution functions can also easily be used by modifying the code.

For simulation of three dimensional fracture networks the following function is used in a similar fashion.

$$polys = GenFNM3D(n, dip, ddip, ddir, dddir, rgn, s);$$

where n is the number of fractures, *dip* is the mean dip angle $[0..\pi/2)$, *ddip* is variation limit around the *dip* angle ($0 \le ddip \le \pi/4$), *ddir* is the mean orientation $[0..2\pi)$ and *dddir* is variation limit around the *ddir* angle ($0 \le dddir \le \pi$) for fractures, *rgn* is the region of study $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$ (a cube) by which the simulated fracture network is clipped and *s* is the scaling factor to determine maximum size (s=S_{max}) for generated fracture lengths which follow negative exponential distribution. The resulting data is of Matlab "cell" datatype in which every element contains coordinates of fracture polygons. Note that, in this implementation a four-vertex fracture polygon (Fadakar-A et al., 2011) would result in an oriented irregular polygon with between 4 and 8 vertices due to clipping by the region (cube) boundaries. Hence the cell structure is preferred to accommodate all variants of number of vertices in the resulting polygons. Referring to the source code of GenFNM3D function, it becomes clear that this function can handle any convex polygon (Corrochano, 2005) with an arbitrary number of vertices including digitized ellipses, with no additional work.

According to Matlab coding convention, the two introduced simulation functions can also be used as [lines, olines] = GenFNM2D(…); and [polys, opolys] = GenFNM3D(…); where *olines* and *opolys* are the simulated fractures without clipping. Example fracture networks for two and three dimensional cases are demonstrated in Fig. 1. Their associated parameters and statistics are *Kappa* = 0 i.e., a Uniform distribution (U) for *theta* bounded to [0,90] degrees; where length follows negative Exponential distribution (E) bounded between 0.05 and 0.5 for two-dimensional model. For three-dimensional models the dip angle varies between 0 and 45 degrees following Fisher distribution around the mean dip, 45 degree, and dip-direction angle varies between 0 and 360 degrees. Simulation of fracture networks by means of ADFNE is quite efficient and fast as it takes only about 23 s for two-dimensional case and only 6 min for three-dimensional case for 100,000 fractures each (tested on a laptop: CPU one core 1.8 GHz, RAM 2 GB). These timings include fracture and network generation, and intersection and clustering analyses, as well.

Adjustment of settings for the function helps to simulated different models with no limits. For example, in Fig. 2 variation in the size constraint (*s*) of a three dimensional fracture network model is demonstrated. These capabilities provide opportunities to represent complicated fracture networks.

The procedural structure of ADFNE and its functions allow simulating even very complex DFN models by combining multiple simulations and stages. For example, in two-dimensional case, one may opt to honor various geological regions with different DFN models. This goal can easily be achieved by means of the parameter *rgn* in GenFNM2D, for example. A simple framework for this would include dividing the region of study into desired sub-regions (polygons) corresponding to the geological settings. The resulting polygons can then be used as the *rgn* value in multiple runs of GenFNM2D. Finally, all generated DFN models are combined (using Matlab built-in functions) to build a single complex DFN (see Fig. 3 for a demonstration). The framework described here is of simple but effective conditional simulation techniques. Similar strategies can be planned for three-dimensional cases. Note that, the clipping shape available in the current release of ADFNE is a cube for three-dimensional fracture networks; nevertheless, the above mentioned steps can be used as guidelines for simulating complicated scenarios.

In similar steps, one may opt to apply inhomogeneous density of fractures over the study area. This goal can easily be achieved by adjusting a proper point process for the locations in both two- and three- dimensional cases. A recommendation here would be to duplicate the generator function (e.g., GenFNM2D) and then modify the locations inside (pts = rand(n,2);) to any desired form. Apparently, the distribution function used (e.g., rand which stands for Uniform distribution) can be replaced by any random function or even combination of multiple functions (e.g., if multiple modes are required). For example, using pts = randn(n, 2); (Matlab built-in function for Gaussian distribution) would result in centrally dense fracture networks useful for especial cases. Similar approaches can be pursued for three-dimensional DFN models. The main finding here is that, complex systems could be somehow satisfactorily modeled by iterative and or multiple runs as well as customized functions. Future releases of ADFNE would hopefully provide even more predefined functions for advanced conditional simulations including honoring borehole and well data, geological settings, mechanical and geomechanical properties, and so on.

3.2.1. Verifying DFN simulation results

In principles, if the amount of available useful information is extensive for DFN simulations, the resulting DFN will be quite satisfactory as it will be more representative of the reality. Generally, cross-validation may be conducted using "constructing" parameters such as length distributions or by means of "responses", such as connectivity, permeability etc. In the next section, both methods are explained in details. Practically speaking, depending on the purpose of simulation, one might choose one method over the other. Characterization tools available in ADFNE help to extract key information from candidate fracture networks (measured vs. measured, measured vs. simulated or simulated vs. simulated) to assess for differences and or to conduct cross-validation. It is also possible to compare responses (secondary characteristic, e.g., connectivity) between fractured networks.

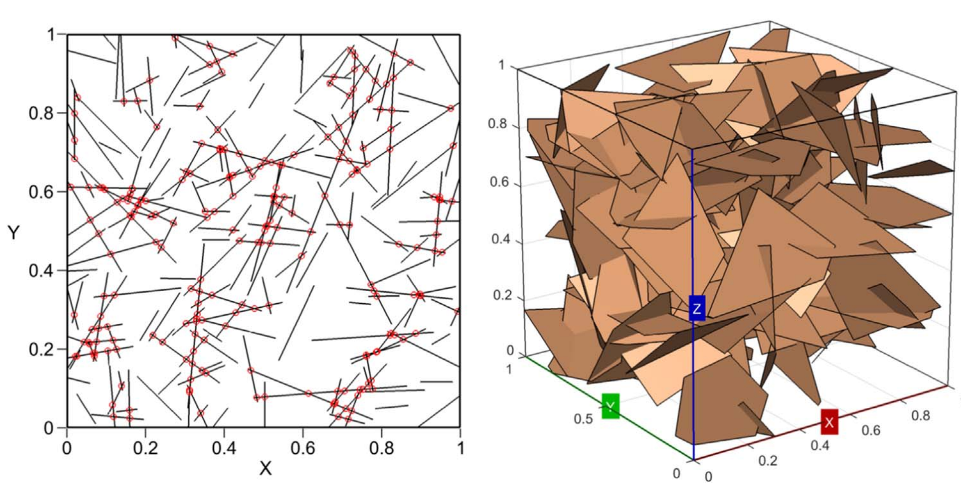

Fig. 1. Simulated two and three dimensional fracture networks. Settings were $\{n = 200,\ theta = U(0,\ 2\pi),\ L = E(0.05,\ 0.5)\}$ and $\{n = 100,\ dip = (0,\ \pi/2),\ ddir = (0,\ 2\pi),\ S_{max} = 0.5\}$, respectively.

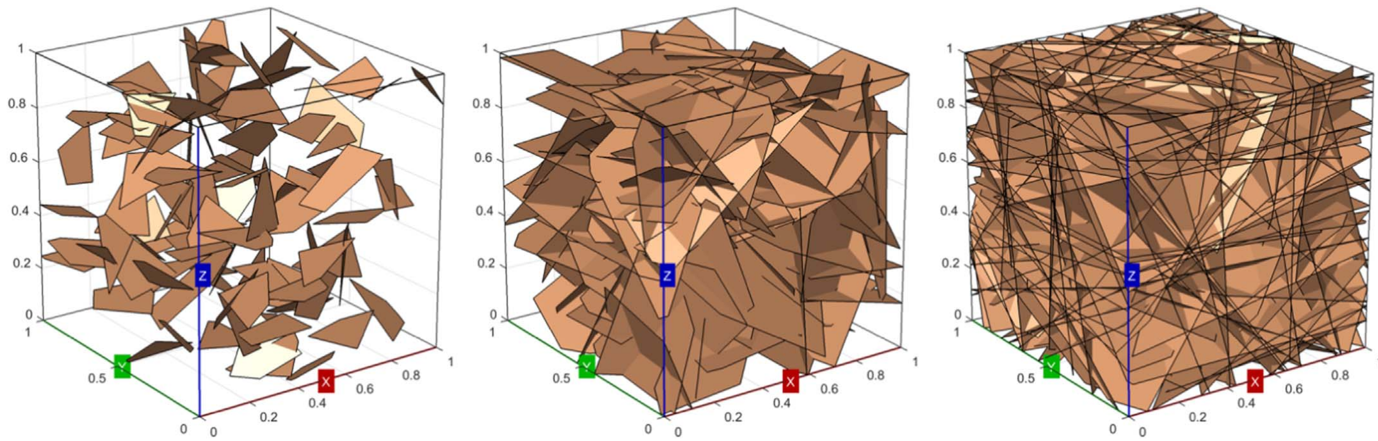

Fig. 2. Simulated three dimensional fracture networks with different fracture maximum size (S_{max}) limits 0.25, 0.75 and 2 from left to right respectively.

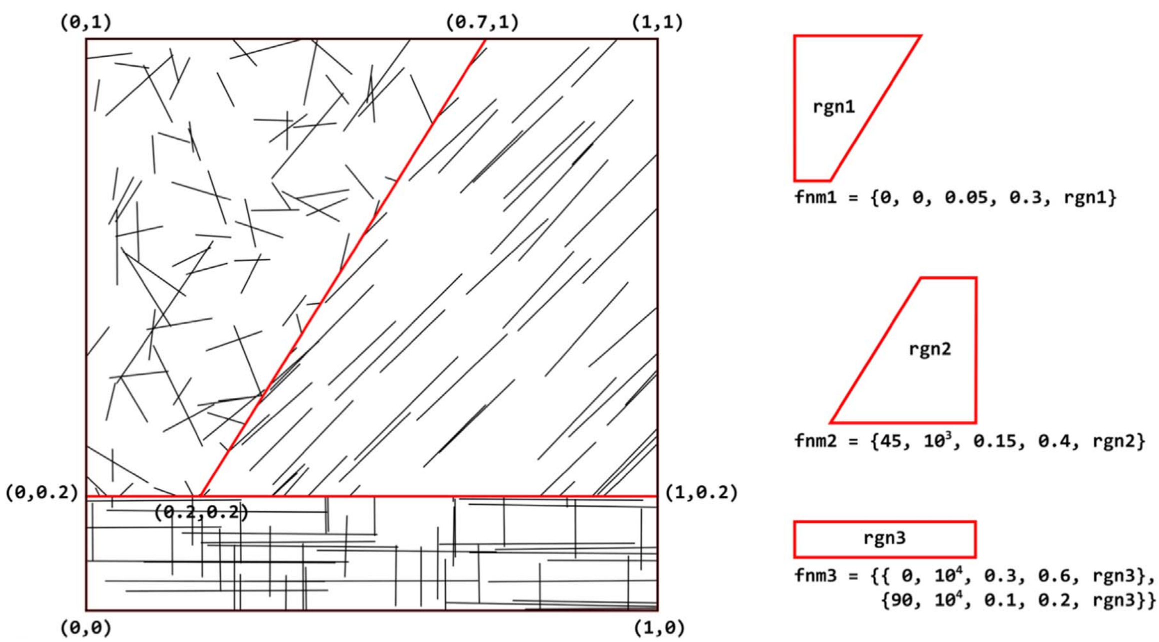

Fig. 3. A complex DFN is built over multiple simpler fracture networks. For this example, the parameters used for GenFNM2D are shown in the right. Region polygons can be any simple or complex shape. The separations can be due to fault or layering, for example.

3.3. Characterization of fracture networks

A fracture network can be a complex phenomenon and difficult for characterization. Geometrical complexity and topological complications (e.g., due to large number of intersecting fractures in the network) make the characterization a challenging problem. It becomes even more complicated and difficult when three dimensional fracture networks are the case. A key requirement here is to establish a framework capable of extracting intersection information for any fracture network in an efficient way. Intersection analysis (Fadakar-A et al., 2011; Fadakar-A, 2014) examines association between fractures. The resulting intersection information (intersection points, indices and so on) leads to determination of any interconnections between fractures, i.e., connectivity of fracture networks (Ozkaya and Mattner, 2003; Fadakar-A et al., 2014). Modeling of flow (fluid or heat) through fractures (Dverstorp, 1991; CFCFF, 1996; Berkowitz, 2002; Karvounis and Jenny, 2011; Fadakar-A et al., 2013b) is under direct effect of the connectivity properties of fracture networks, as is the modeling of rock block stability (Goodman, 1985, Jimenez-Rodriguez and Sitar, 2008), for example.

3.3.1. Intersection analysis

In two dimensional fracture networks any two fractures can intersect in a single point (red circles in Fig. 1 left) while for three dimensional fracture networks the resulting intersection could be a point (touching vertex) or line (Fadakar-A, 2014). Either case, intersections are the most important characteristic of fracture networks as they define the connectivity properties of the network which is a key and determinative player in almost any application of DFNE. For example, fluids flow through fractures as they commute from one to another location (fracture) via intersections between fractures; and the movement of rock blocks is mainly associated with the type of connectivity established between fractures (joints).

Analysis of intersection for two dimensional fracture networks is handled by means of LinesX2D function in ADFNE package. Applying to the fracture network given in Fig. 1(left) as [xts, ids, La] = LinesX2D(lines); results in a total of 224 intersection points marked as red circles. In Fig. 4(right) the contour map shows the density of the intersection points which is also an important factor in the determination of flow distribution in the fracture network. Density of fractures (i.e., true density) shown in Fig. 4(left) is obtained by means of

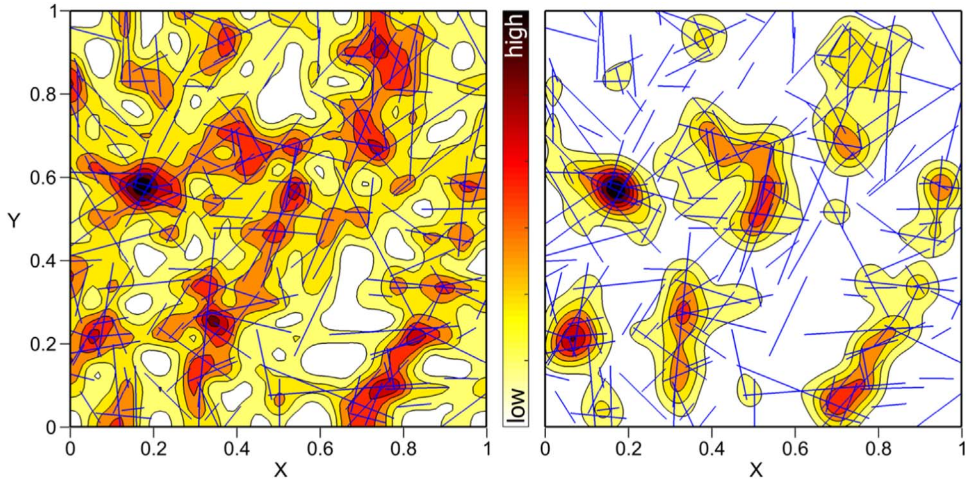

Fig. 4. Density maps; (left) density of fractures, (right) density of intersection points. Compare the significant differences, top-right corner, for example. Both maps report densities projected between 0 (low) and 1 (high).

Density2D function (see also Section 3.4). The comparison given in Fig. 4 is to emphasize the differences between the two densities purely in terms of overall trend and distribution (more comparisons in Fadakar-A et al., 2011). Note that, as the two density measures apparently vary in different domains to make the comparison meaningful, focused and unbiased we projected both densities into [0..1] corresponding to the low and high areas in the figure. It is worth noting here that any result in Matlab and so in ADFNE is ultimately a collection of numbers. That is, for interested user all values are always accessible. With that being said, for the purpose of this comparison, i.e., the spatial trend of variation, observing the areas with higher density of fractures but lower density of intersections (e.g., top-right) depicts the fact that the density of fractures does not necessarily correspond to the interconnection between fractures. This is a key parameter in many applications such as modeling of fluid flow through fractures. Note that despite how dense a fracture network would be its role in fluid flowing (under some assumptions such as channeling effect and governing simple relationships between hydraulic properties and fracture surface geometry) is closely affected by the density of intersections. The density of fractures is also suggested to be effective in geo-hazards. Displacement of rock blocks and formations underground under governing stress regime would easier happen if the areas with high density of fractures show high interconnectivity. A closely associated topic in rock engineering is the rock-bridge concept (e.g., in rock slope stability analysis). Hence the two densities can be seen of critical importance for risk evaluations, as well.

Application of intersection analysis to three dimensional fracture networks is also exemplified in Fig. 5 in which the intersection lines are shown in red tubes. The following function extracts complete intersection information from any three dimensional fracture networks.

$$[xts,\ ids,\ La] = PolysX3D(polys);\ \%\ Intersection\ Analysis,\ 3D$$

where *xts* is set of intersection lines, *ids* is set of two intersecting fractures for every line in *xts*, and *La* is the clustering labels for all fractures in the network. In Fig. 5 the histogram of intersection lines (Fadakar-A et al., 2011) is approximately fitted by an Exponential distribution function. This finding appears important in the characterization of fluid flow through fractures as it demonstrates that the flow can pass mainly through small connections between fractures.

3.3.2. Cluster analysis

The resulting information from intersection analysis can be used to determine fracture clusters in the network. A fracture cluster is basically defined as group of interconnected fractures. The interconnection can occur either directly or via intermediate fractures between any two fractures in the network. The clustering information helps to localize the evaluation of the network, as all fractures in a cluster are connected to each other directly or indirectly. This implies observing clustered behaviors (such as the forms seen in fluid flow distribution) in the network in some extent. A connection between two isolated clusters can be made by pairing only two fractures one from each. This suggests an interesting application where expansion of fracture networks is of demand, e.g., in geothermal heat exchange chamber (Hayashi et al., 1999). That is, there are critical areas in the fracture network domain in which if additional fracturing happens larger clusters would emerge. In ADFNE, the clustering is determined by means of the following generic functions.

$$La = Labels(Clusters(ids),\ n);$$

where *ids* is set of two intersecting fractures to which clustering function, Cluster, is applied, function Labels assigns unique labels for every cluster, and *La* is the resulting cluster labels. Note that Labels and Cluster are generic functions and so apply to intersection information obtained for both two and three dimensional cases. Fig. 6 demonstrates an application of clustering to the two and three dimensional fracture networks given in Fig. 1.

3.3.3. Connectivity analysis

Connectivity of fractures in fracture networks is another key characteristic for variety of applications such as stability of rock blocks, and preferential pathways for fluid flow (Priest, 1993, Fadakar-A et al. 2013b-2013c). More connected fractures more the readiness is for slipping and failure of rock mass according to probability principles apply to risk analysis. The complexity of preferential fluid pathways through fractures, spatial distribution pattern of active fracture clusters in the network, approaching percolation threshold, establishing a sustainable flow rate between injection and production wells (i.e., in geothermal and oil reservoir systems) are all closely associated with the connectivity properties of fracture network (Meyer and Einstein, 2002; Xu et al., 2006; Fadakar-A et al., 2014; Preisig et al., 2015). Hence, the evaluation of connectivity and understanding its behavior in fracture networks is a fundamental study which requires a proper definition and comprehensive framework, indeed.

– Connectivity Index

A simple connectivity assessment can be carried out by means of evaluation of the probability of connection between two points in an n-dimensional space. Such a probability can be determined via connectivity index (CI, Xu et al., 2006), for example. If two fractures are

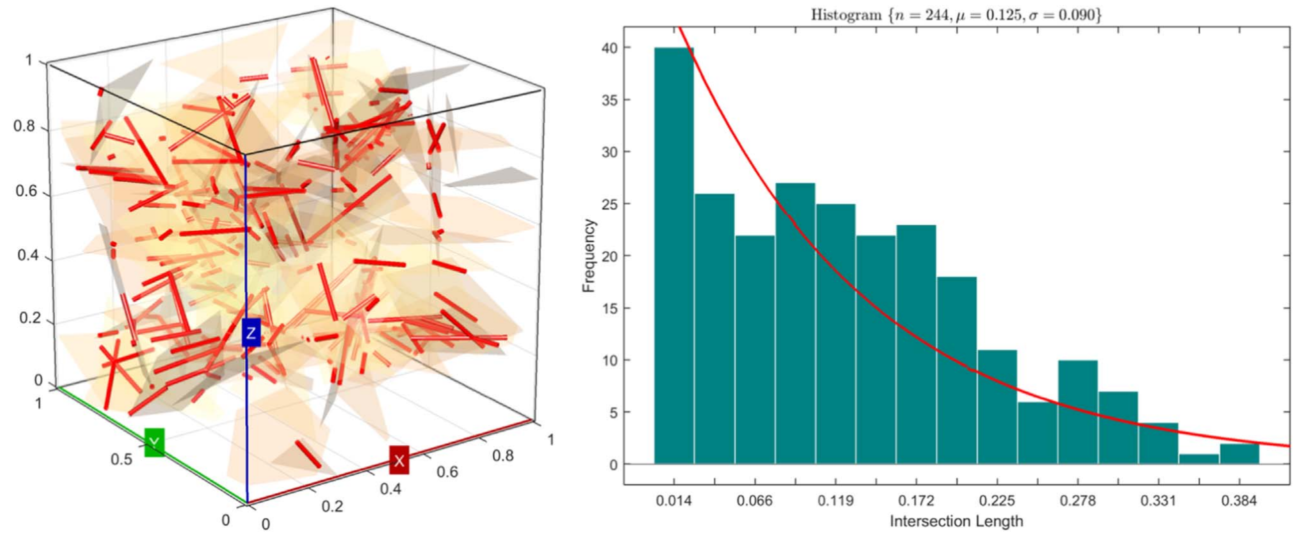

Fig. 5. Intersection analysis applied to the fracture network shown in Fig. 1(right) resulted in 244 intersection lines for which the histogram (right) is drawn with an approximate fit of Exponential distribution (red curve). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

connected directly or via other intermediate fractures, its connectivity value is "one" otherwise "zero", i.e., $\{1: f_1 \leftrightarrow f_2, 0: f_1 \nleftrightarrow f_2\}$. In a simulation of several realizations from a fracture network model the number of times that the two portions (sub-regions, supports) of the area of study remain connected via fractures determines the CI probability value between them. For two dimensional fracture networks the CI can be computed as follows.

$$CI = ConnectivityIndex2D(lines,\ La,\ d1,\ d2,\ p,\ q)$$

where *lines* is fracture network, *La* is cluster labels, *d1* and *d2* are dimensions for the sampling grid, *p* and *q* are indices for target cell, and *CI* is the resulting connectivity index. In Fig. 7 the CI is applied to two sets of 60 realizations obtained from simulation of two fracture network models. The first model is omnidirectional, while the second is mainly (*kappa*=10) oriented towards 45 degrees counterclockwise.

– Connectivity Field

The connectivity evaluation for the entire fracture network is obtained by means of the Connectivity Field (CF, Fadakar-A et al., 2014) measure. Practically speaking, in a simplistic implementation, the whole study area is mapped onto a grid in which for any pair of cells the connectivity is measured. The evaluation can be further developed by incorporation of weighting system (Fadakar-A et al., 2013b) to distinguish individual connectivity elements due to length, aperture and permeability, for example. Obviously, the CF applies same principles to two- and three-dimensional fracture networks. The following function computes the CF for two dimensional cases.

$$CF = ConnectivityField2D(lines,\ La,\ d1,\ d2)$$

where *CF* is the resulting connectivity field. In Fig. 8 CF is applied to the two dimensional fracture network model given in Fig. 1.

The large grid cell sizes (i.e., smaller *d1* and *d2* values) would cause the CF to appear blocky, Fig. 8 (left). On the other hand, very small cell sizes could extract no further information than fracture itself. A solution is to implement the Generalized CF (GCF, Fadakar-A et al., 2014), as shown in Fig. 8 (right). The GCF can be computed by a loop over CF with increasing *d1* and *d2* values. The resulting individual CF matrices can then be resized by means of Resize2D function into a fixed size. The mean value of all matrices produces the GCF, i.e., an "E-type" map. The GCF is reported to be robust against cell size choice as well as clean of edge effects due to sampling grid (see Fadakar-A et al., 2014 for detailed discussions).

3.4. Density analysis

Traditional measures for quantifying density (and intensity) of

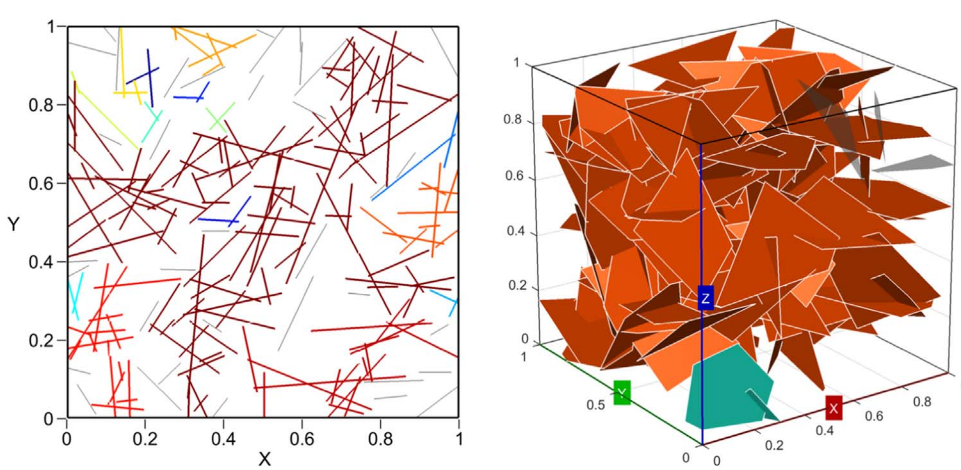

Fig. 6. Clustering applied to the same two and three dimensional fracture networks introduced in Fig. 1. Isolated fractures are shown in gray. Warmer the color, larger the cluster size (i.e., the number of members) is. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

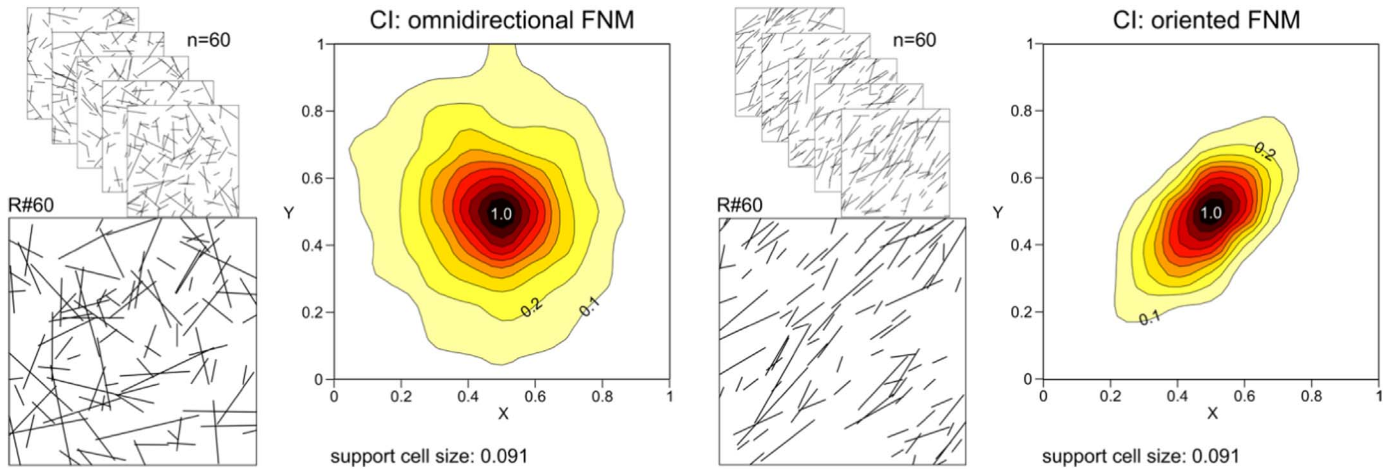

Fig. 7. Connectivity Index for two dimensional fracture network models. (left) CI for omnidirectional model, (right) CI for oriented model. Apparently, CI is affected by the preferential orientation of fractures in the network.

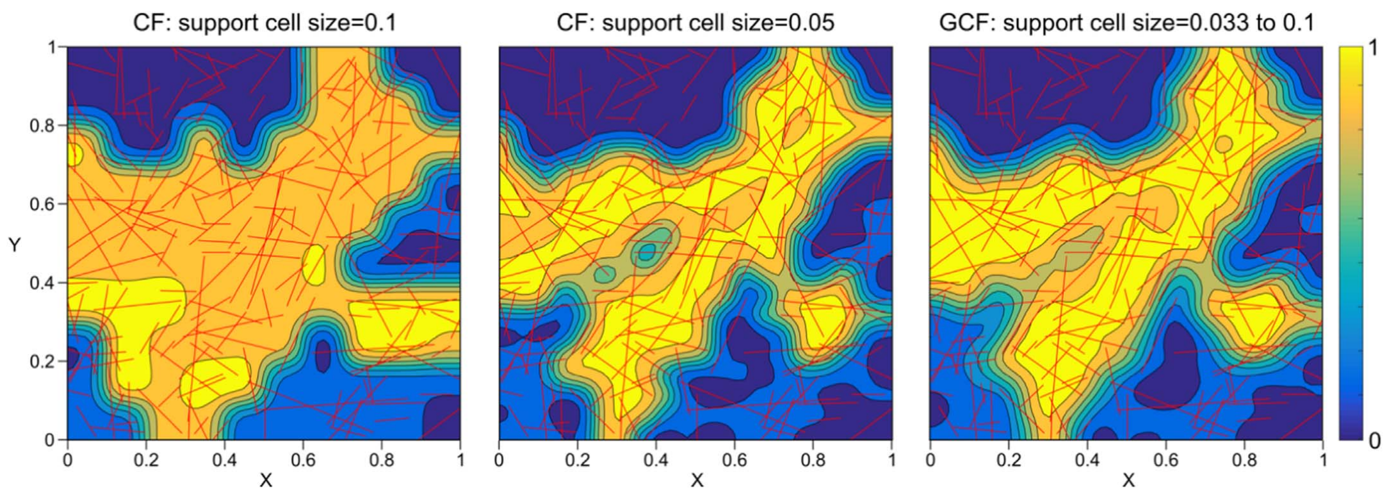

Fig. 8. Connectivity Field for the two dimensional fracture network given in Fig. 1. (left) CF computed on sampling grid (10×10); (middle) CF computed on sampling grid (20×20); (right) Generalized CF, sampling grid (10×10..30×30).

fracture networks include P series from which P21 (length of fractures per area) and P32 (area of fractures per volume) are common (Dershowitz et al., 2000). These measures are easy to be quantified, but, produce only a single value for the entire network. Fracture networks are often heterogeneous in many aspects including the density, hence, a single value measures appear too simplified. The solution is to use a comprehensive and advanced measure, the Fracture Density (FD, Fadakar-A, 2014, for example), in which the measurement is carried out on an arbitrary grid covering the study area. Every cell is examined for the number of intersecting fractures. The total length (area in three-dimension cases) of fracture parts can also be taken into account for determination of the fracture density in cells. The resulting is a complete density map for the fracture network. The FD can be extended into three-dimension under the same concept and straightforwardly. The FD can also be seen as the field of density (i.e., full information). The density maps can be generated as follows.

$$FD = Density2D(lines, La, m, n)$$

where *FD* is the resulting density field (map). Fig. 9 demonstrates the FD applied to a two dimensional fracture network given in Fig. 1.

An interesting and important note here is that, the computation of FD is neither constrained to a specific cell shape (e.g., square) nor to sampling grid. Indeed, the principles are the same if one chooses an arbitrary shape for cells (e.g., circle) and or an arbitrary sampling scheme (e.g., random sampling). This is a key feature of the concept and can be implemented by means of ADFNE functions quite straightforwardly. For example, intersections between fractures and a circle can be found by digitizing a circle as a polygon. There are also opportunities to implement analytical solutions, if preferred.

3.5. Graph theory

Application of graph theory would benefit many operations in characterization of fracture networks and would extend their uses significantly (Fadakar-A et al., 2013b). A rigorous framework borrowed from principles of graph theory includes extraction of nodes, edges and their associations (topology). The first step here is to extract the backbone (also known as skeleton, Priest, 1993) structure of fracture networks which is by itself a key step for fluid flow modeling in the network, for example. Generating backbone from two- and three-dimensional fracture networks follows the same stages as easy as follows.

$$\begin{aligned}
bls &= BreakLinesX2D(lines); \;\% \text{ 2D FNM} \\
bbn &= Backbone(bls); \;\% -- \text{ Backbone} \\
pip &= Pipes3D(polys); \;\% \text{ 3D FNM} \\
bbn &= Backbone(pip); \;\% -- \text{ Backbone}
\end{aligned}$$

where *lines* is fracture network, *bls* is broken lines in their intersections by means of BreakLinesX2D function, and *bbn* is the resulting backbone structure. For three dimensional fracture network (*polys*), Pipes3D function generates pipe model from which the backbone structure is

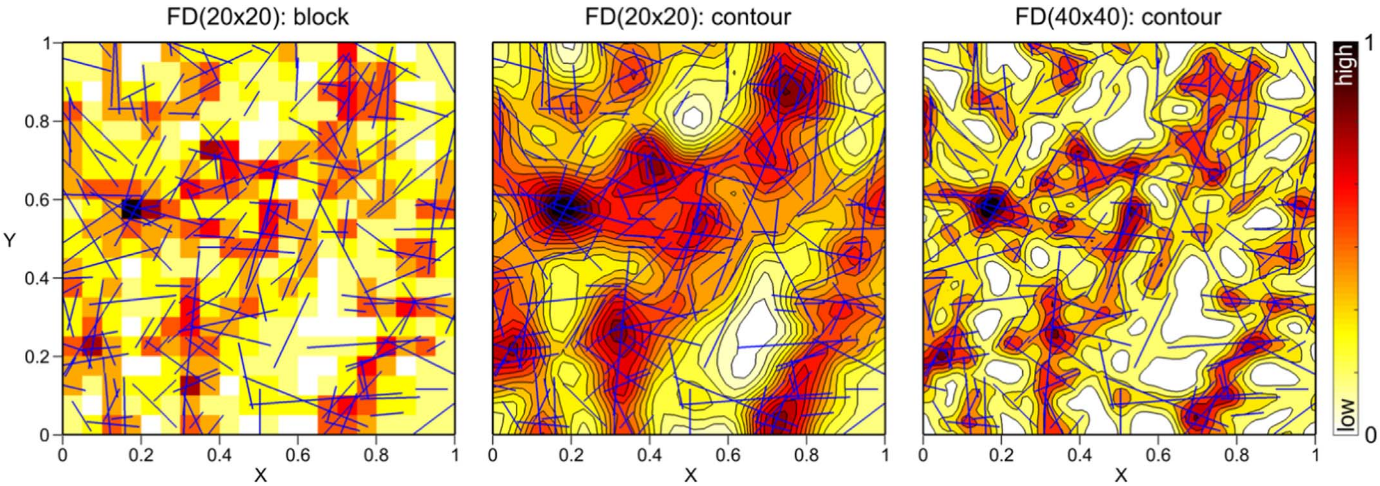

Fig. 9. The Density Field for the two dimensional fracture network given in Fig. 1. (left) The block map of the FD computed on the sampling grid of 20×20; (middle) The contour map of the CF computed on the sampling grid of 20×20; (right) The contour map of the FD computed on the sampling grid of 40×40.

extracted. A pipe for a fracture can be made between its centroid and its center of intersection with another fracture. Fig. 10 demonstrates backbone structures extracted from the two- and three-dimensional fracture networks given in Fig. 1.

Generic function BackboneToGraph transforms backbone structure into graph structure in which nodes, edges and the associated topology provide an exceptional opportunity to investigate complex properties of fracture networks in a very straightforward manner. An example for this is the fluid flow modeling (Fadakar-A et al., 2013b) by application of finite difference methods which can be elegantly done based on the graph inlet, inner and outlet nodes and some efficient functions such as neighboring.

3.6. Flow modeling

A simple and handful fluid flow modeling technique is founded on an application of finite difference method (Priest, 1993, Fadakar-A et al., 2013b). The basics are quite simple, well studied and documented. All income and outgoing flow to and from a node must fully match, that is, mass conservation is in place. Under Darcy's law for laminar flow of incompressible fluid, the following simplified equations are used to determine the pressure head distribution for every node in the graph of any two-dimensional fracture networks (Priest, 1993).

$$H_j = \frac{\sum_{i=1}^{n} C_{ij} H_i}{\sum_{i=1}^{n} C_{ij}}, \qquad n \in [2, 3, 4]; \qquad C_{ij} = \frac{ga^3 b}{12vL}$$

where C_{ij} is the conductance between nodes i and j calculated based on g, the gravity acceleration (9.8 $\mathrm{m/s^2}$), a, the aperture (m), b, the third dimension of fracture (here 1 for two-dimensional case), v, the kinematic viscosity ($10^{-6}\ \mathrm{m^2/s}$ for water), and L, the length (m). H_j is the head pressure at node j.

Practically, nodes are categorized into three types: inlet, inner and outlet nodes. The flow (here pressure head, for example) starts from the inlet nodes and distributes through the inner nodes to reach the outlet nodes. Based on graph structure, for every inner node the neighboring nodes are found for which the matrix of unknowns is constructed following the above equations. As shown, the values inserted in the matrix incorporate conductance factor (Priest, 1993) which itself is calculated based on the neighboring edges' properties as explained. When the matrix system (**A**X=B) is fully prepared, a simple linear algebra technique helps to find a solution (X=**A**/B). The resulting solution includes pressure head value for all inner nodes. The following functions from ADFNE help to achieve the mentioned flow solution.

$g = BackboneToGraph(bbn);$
$g = SolveFlow(g,\ inlet,\ outlet);$

where *bbn* is backbone, g is the resulting graph. SolveFlow function accepts the graph g, *inlet* and *outlet* pressure heads and returns the solution as an updated graph.

In Fig. 11 an example fluid pressure flow problem is solved as described. The resulting pressure head values at nodes and the interpolated pressure values for edges are demonstrated in Fig. 11

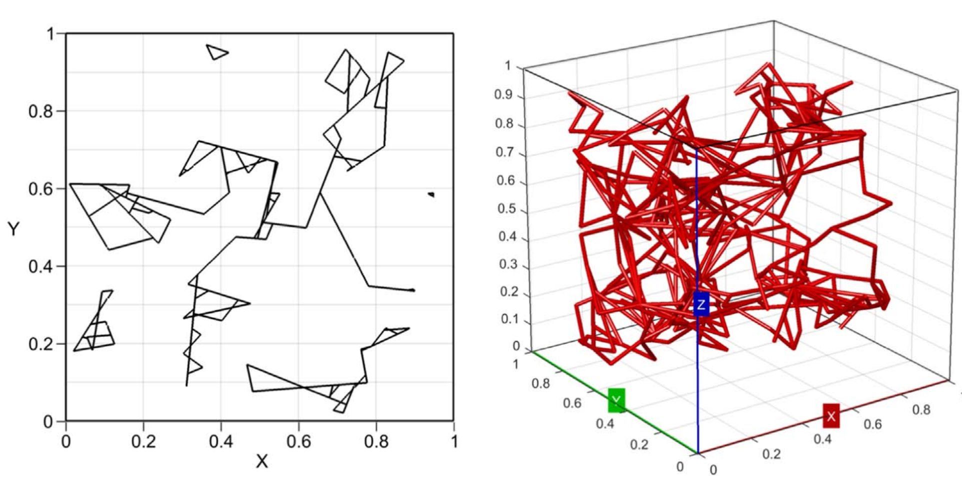

Fig. 10. Backbone (Skeleton) structure of the two- and three-dimensional fracture networks given in Fig. 1.

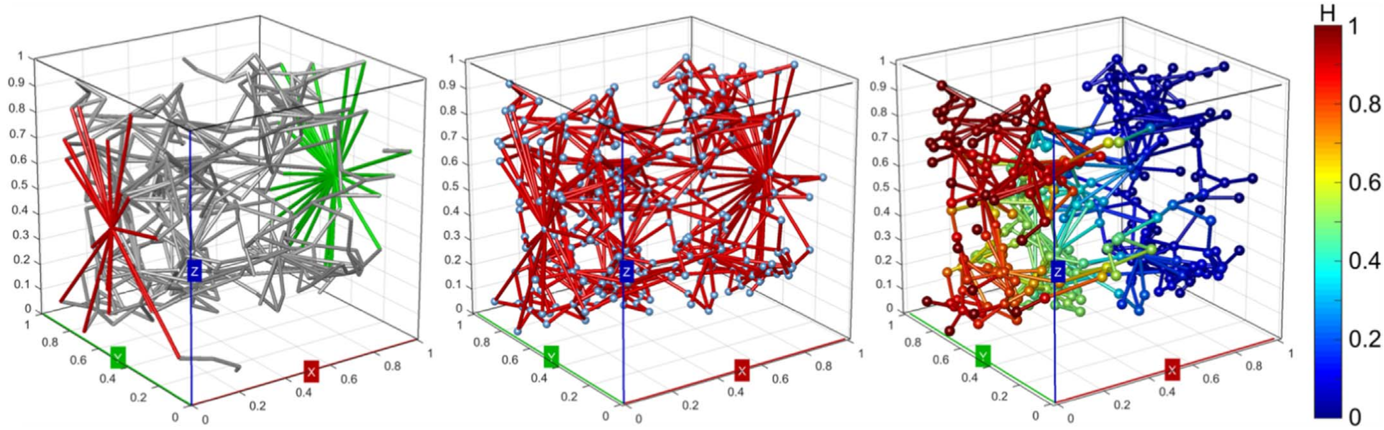

Fig. 11. Modeling of fluid flow (pressure head) through fracture network model given in Fig. 1. (left) Added two inlet and outlet polygons generate feeding pipes. (middle) graph structure (nodes and edges), (right) the solution for pressure head in which inlet (H=1) and outlet (H=0) nodes are in dark red and dark blue, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

depicting the direction of pressure diffusions. The flow modeling demonstrated above assumes channeling in effect on fracture networks. Due to this effect the fluid commutes through preferential pathways on fracture surfaces connecting one to another in the network simplified here as the backbone structure.

4. Future developments

The diversity of DFNE applications and users' needs require continuous development of ADFNE package. This demand is further emphasized by recent exceptional growth of DFNE concepts both in academia and industry. Contributions of the community of DFNE in the future developments of ADFNE package would include adding *dedicated* functions for conducting (i) comprehensive conditional simulations (e.g., honoring borehole data, logs, and density map, etc.), (ii) complete flow simulations (e.g., modeling permeability of rock mass, fluid and or heat flow, pressure, etc.), (iii) geomechanically coupled fracture simulations (e.g., adapted truncation modes), and (iv) high performance simulations using parallel and cluster computing. It would also be useful to have additional functions to conduct spatial and statistical analysis on fracture and cluster drainage area (volume).

5. Conclusions

An open source software package, Alghalandis Discrete Fracture Network Engineering, ADFNE, was introduced with example codes and demonstrations. A quick review of fundamentals of DFNE was also provided. It was shown that several key topics in DFNE including fracture generations, simulation, characterization (intersection analysis, connectivity index, connectivity field, density and so on), applications such as evaluation of the risks in rock mass associated with the density of fractures and fracture intersections, the fluid flow modeling are covered in ADFNE. The simplistic end-user interface (i.e., handy, generic and flexible functions) but efficient and fairly sophisticated internal design and advanced coding provide reliable framework for conducting researches and even future developments. All the illustrations in this paper were produced by ADFNE, as well as the development of the concepts. The package is aimed to help and encourage primarily researchers in the field of DFNE for exploring, examining and quick implementation of ideas but would even be useful in any other disciplines. The current release of ADFNE package together with sample complete programs and tutorials are available for download at http://alghalandis.net/products/adfne.

Acknowledgements

We thank you anonymous reviewers for their in depth comments and precious suggestions.

References

Andersson, J., Dverstorp, N., 1987. Conditional simulations of fluid flow in three-dimensional networks of discrete fractures. J. Water Resour. Res. 23 (10), 1876–1886.
Baddeley, A., 2010. Analysing Spatial Point Patterns In R. Workshop Notes. CSIRO, Australia, 232.
Berkowitz, B., 2002. Characterizing flow and transport in fractured geological media: a review. J. Adv. Water Resour. 25, 861–884.
Blocher, M.G., Cacace, M., Lewerenz, B., Zimmermann, G., 2010. Three dimensional modelling of fractured and faulted reservoirs: framework and implementation. J. Chem. Der Erde 70 (S3), 145–153.
CFCFF, 1996. Rock Fractures and Fluid Flow: Contemporary Understanding and Applications. National Academy Press, 568.
Chiles, J.P., 2004. Stochastic modeling of natural fractured media: a review. In: Leuangthong, O., Deutsch, C.V. (Eds.), Geostatistical Banff, 285–294.
Corrochano, E.B., 2005. Handbook of Geometric Computing, Applications in Pattern Recognition, Computer VisionNeural Computing, and Robotics. Springer, 774.
Cosgrove, J.W., 1998. The role of structural geology in reservoir characterization. In: Coward, M.P., Daltaban, T.S., Johnson, H. (Eds.), Structural Geology in Reservoir Characterisation 1–13. Geological Society, Special Publication, 127.
Dershowitz, W.S., La-Pointe, P.R., Doe, T.W., 2000. Advances in discrete fracture network modeling. In: Proceeding of the Us EPA/NGWA Fractured Rock Conference, pp 882–894.
Deutsch, C.V., Cockerham, P.W., 1994. Practical considerations in the application of simulated annealing to stochastic simulation. Math. Geol. 26, 67–82.
Diggle, P., 2003. Statistical Analysis of Spatial Point Patterns. Edward Arnold Publication, 159.
Dverstorp, B., 1991. Analyzing flow and transport in fractured rock using the discrete fracture network concept. R. Inst. Technol. Stockh. BN:TRITA-VBI 151, 64.
Elmo, D., Stead, D., Eberhardt, E., Vyazmensky, A., 2013. Applications of finite/discrete element modelling to rock engineering problems. Int. J. Geomech. 13 (5), 565–580.
Elmo, D., Rogers, S., Stead, D., Eberhardt, E., 2014. Discrete Fracture Network approach to characterise rock mass fragmentation and implications for geomechanical upscaling. Min. Technol.: Trans. Inst. Min. Metall. Sect. A 123 (3), 149–161.
Fadakar-A, Y., 2014. Stochastic Modelling of Fractures in Rock Masses 349. the University of Adelaide, Australia, (PhD Thesis) ⟨http://hdl.handle.net/2440/92338⟩.
Fadakar-A, Y., 2016. Alghalandis discrete fracture network engineering (ADFNE) package. Alghalandis Computing ⟨http://alghalandis.net/products/adfne⟩.
Fadakar-A, Y., Xu, C., Dowd, P.A., 2011. A general framework for fracture intersection analysis: algorithms and practical applications. In: Proceedings of Australian Geothermal Energy Conference AGEC2011, Melbourne, Australia, pp 15–20.
Fadakar-A, Y., Dowd, P.A., Xu, C., 2013a. The RANSAC method for generating fracture networks from micro-seismic event data. J. Math. Geosci. 45 (2), 207–224.
Fadakar-A, Y., Dowd, P.A., Xu, C., 2013b. A connectivity-graph approach to optimising well locations in geothermal reservoirs. In: Proceedings of Australian Geothermal Energy Conference AGEC2013, Brisbane, Australia, pp 111–115.
Fadakar-A, Y., Xu, C., Dowd, P.A., 2013b. Connectivity index and connectivity field towards fluid flow in fracture-based geothermal reservoirsProceedings of 38th Workshop on Geothermal Reservoir Engineering. Stanford University, Stanford,

California, 417–427.
Fadakar-A, Y., Dowd, P.A., Xu, C., 2014. Connectivity field: a measure for characterising fracture networks. J. Math. Geosci. 47 (1), 63–83.
Follin, S., Stigsson, M., Leven, J., 2006. Discrete fracture network characterization and modeling in the Swedish program for nuclear waste disposal in crystalline rocks using information acquired by difference flow logging and borehole wall image logging. Am. Geophys. Union H12, A–04.
Freeden, W., Zuhair-N, M., Sonar, T., 2010. Handbook of Geomathematics. Springer, 1338.
Goodman, J.E., O'Rourke, J., 2004. Handbook of Discrete and Computational Geometry. Chapman & Hall/CRC, 1558.
Goodman, R.E., Shi, G., 1985. Block Theory and its Applications to Rock Engineering. Prentice-Hall International, London, 338.
Grasby, S.E., Allen, D.M., Bell, S., Chen, Z., Ferguson, G., Jessop, A., Kelman, M., Ko, M., Majorowicz, J., Moore, M., Raymond, J., Therrien, R., 2012. Geothermal Energy Resource Potential of CanadaOpen File 6914. Geological Survey of Canada, 322.
Gringarten, E., 1996. 3-D geometric description of fractured reservoirs. J. Math. Geol. 28, 881–893.
Hanano, M., 2004. Contribution of fractures to formation and production of geothermal resources. Renew. Sustain. Energy Rev. 8, 223–236.
Hayashi, K., Willis-Richards, J., Hopkirk, R.J., Niibori, Y., 1999. Numerical models of HDR geothermal reservoirs-a review of current thinking and progress. J. Geotherm. 28, 507–518.
Hernqvist, L., 2009. Characterization of the Fracture System in Hard Rock for Tunnel GroutingLicentiate Thesis. Chalmers University of Technology, Sweden.
Huseby, O., Thovert, J.F., Adler, P.M., 1997. Geometry and topology of fracture systems. J. Phys. 30, 1415–1444.
Illian, J., Penttinen, A., Stoyan, H., Stoyan, D., 2008. Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley & Sons, England, 557.
Jimenez-Rodriguez, R., Sitar, N., 2008. Influence of stochastic discontinuity network parameters on the formation of removable blocks in rock slopes. J. Rock. Mech. Rock. Eng. 41 (4), 563–585.
Jing, L., 2003. A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering. Int. J. Rock. Mech. Min. Sci. 40, 283–353.
Jing, L., Stephansson, O., 2007. Fundamentals of Discrete Element Methods for Rock Engineering Theory and Applications. Elsevier, 545.
Karvounis, D., Jenny, P., 2011. Modeling of flow and transport in enhanced geothermal systems. In: Proceedings of 36th Workshop on Geothermal Energy Systems, Stanford, California, USA.
Kendall, W.S., 2003. Models and Simulation Techniques from Stochastic Geometry, a course of Madison Probability Intern Program. The University of Warwick, 163.
Koike, K., Ichikawa, Y., 2006. Spatial correlation structures of fracture systems for deriving a scaling law and modeling fracture distributions. J. Comput. Geosci. 32, 1079–1095.
Koike, K., Kubo, T., Liu, C., Masoud, A., Amano, K., Kurihara, A., 2015. 3D geostatistical modeling of fracture system in a granitic massif to characterize hydraulic properties and fracture distribution. J. Tectonophys. 660, 1–16.
Koyama, T., Li, B., Jiang, Y., Jing, L., 2009. Numerical modelling of fluid flow tests in a rock fracture with a special algorithm for contact areas. J. Comput. Geotech. 36, 291–303.
Mathworks, 2016. Matlab Product Documentation. ⟨http://mathworks.com⟩.
Merrien-Soukatchoff, V., Korini, T., Thoraval, A., 2012. Use of an integrated discrete fracture network code for stochastic stability analyses of fractured rock masses. J. Rock. Mech. Rock. Eng. 45, 159–181.
Meyer, T., Einstein, H.H., 2002. Geologic stochastic modeling and connectivity assessment of fracture systems in the Boston area. J. Rock. Mech. Rock. Eng. 35 (1), 23–44.
Nelson, R.A., 2001. Geologic analysis of naturally fractured reservoirs. Gulf Prof. Publ., 350.
Ozkaya, S.I., Mattner, J., 2003. Fracture connectivity from fracture intersections in borehole image logs. J. Comput. Geosci. 29, 143–153.
Preisig, G., Eberhardt, E., Gischig, V., Roche, V., Van Der Baan, M., Valley, B., Kaiser, P.K., Duff, D., Lowther, R., 2015. Development of connected permeability in massive crystalline rocks through hydraulic fracture propagation and shearing accompanying fluid injection. . J. Geofluids 15 (1–2), 321–337.
Priest, S.D., 1993. Discontinuity Analysis for Rock Engineering. Chapman & Hall.
Python Foundation, 2016. Python Lang. ⟨http://python.org⟩.
Singhal, B.B., Gupta, R.P., 2010. Applied Hydrogeology of Fractured Rocks. Springer, 408.
Staub, I., Fredriksson, A., Outters, N., 2002. Strategy for a rock mechanics site descriptive model. Golder Assoc., Swed., 219.
Tang, C., Kaiser, P.K., Yang, G., 1996. Numerical simulation of seismicity in rock failure. In: Aubertin, M., Hassani, F., Mitri, H. (Eds.), Rock Mechanics: Tools and Techniques 2, 1833–1840.
Vince, J., 2005. Geometry for Computer Graphics. Springer, London, 359.
Vogel, H., 2002. Topological characterization of porous media. In: Mecke, K.R., Stoyan, D. (Eds.), LNP600. Springer, 75–92.
Wyborn, D., De-Graaf, L., Hann, S., 2005. Enhanced geothermal development in the Cooper Basin area, South Australia. Trans., Geotherm. Resour. Counc. 29, 151–156.
Xu, C., Dowd, P.A., Mardia, K.V., Fowell, R.J., 2006. A connectivity index for discrete fracture networks. J. Math. Geol. 38 (5), 611–634.
Zimmerman, R.W., Bodvarsson, G.S., 1996. Hydraulic conductivity of rock fractures. Transp. Porous Media 23, 30.