Stochastic Modelling *of* Fractures *in* Rock Masses

Younes Fadakar Alghalandis

A thesis submitted for the degree of

Doctor of Philosophy

School of Civil, Environmental and Mining Engineering

Faculty of Engineering, Computer and Mathematical Sciences

The University of Adelaide

THE UNIVERSITY *of* ADELAIDE

March 2014

Dedication

to my mother and my father

Contents

Table of figures

Abstract

Fracture and fracture network modelling is a multi-disciplinary research area. Although the literature in general is significant, many research challenges remain. The complex geometry and topology of realistic fracture networks largely determine the static and dynamic mechanical properties of rock. In applications to hot dry rock geothermal reservoirs it is not possible to observe or measure fractures directly on any scale and the only data available are indirect measurements, such as seismic activity generated by hydraulic fracture stimulation. The lack of direct data and the complexities of the fracture characteristics make fracture network prediction and modelling in these applications very difficult. The ultimate purpose of the fracture and fracture network models is to evaluate the response of the fracture system to stress regimes and fluid flow. As understanding of the effective factors in the geometrical modelling of fractures and consequently topological properties of fracture networks increases, more accurate and hence more reliable results can be achieved from associated analyses. For flow modelling in geothermal reservoirs, the critical component of a fracture model is the connectivity of the fractures as this determines the technical feasibility of heat production and is the single most significant factor in converting a heat resource to a reserve. The ability to model this component effectively and to understand the associated system is severely constrained by the lack of direct data. In simulations, the connectivity of a fracture network can be controlled to a limited extent by adjusting the fracture and fracture network parameters (e.g., locations, orientations) of the defining distribution functions. In practical applications connectivity is a response of the system not a variable. It is essential to pursue modelling methods that maximise the extraction of information from the available data so as to achieve the highest possible accuracy in the modelling. Although the evaluation of fracture connectivity is an active research area, widely reported in the literature, almost all connectivity measures are based on degraded representations of the fracture network i.e., lattice-based. The loss of fracture connectivity information caused by using discrete representations is significant even when very high resolutions (assuming they are feasible) are used. This is basically due to the fact that the aperture dimensions of

fractures are several magnitudes smaller than their lengths. If discretisation is necessary, then a better approach would be to retain all connectivity information between fractures, i.e. for connectivity information to remain invariant to the resolution of the discretisation. Such a method would provide more reliable evaluation of connectivity. This thesis covers the modelling of fracture networks, the characterisation (particularly connectivity) of fracture networks and applications.

Statement of Originality/Consent/Copyright

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines.

Younes Fadakar Alghalandis

10 March 2014

Acknowledgments

I am thankful to my parents for their lifetime love and support; to my brother and sisters for their limitless kindness, support and encouragement; to my Iranian old friends; to my new friends worldwide; and to whom encouraged me to continue my study.

I would like to thank Professor Peter Dowd for his continuous support as the principal supervisor and Associate Professor Chaoshui Xu as the co-supervisor during my PhD study in the University of Adelaide. They were helpful in many aspects of my research. I am particularly in debt to Professor Dowd editing assistance especially on published papers. Without his help publication in high standard journals would not have happened.

I am also grateful for the allocation of a PhD scholarship funded by the Australian Research Council as part of ARC Discovery Project DP110104766 (Stochastic modelling of fractures in crystalline rock masses for hot dry rock enhanced geothermal systems) awarded to my supervisors.

Finally I am grateful to the University of Adelaide for admission to make this study possible.

1

Introduction

1.1 Stochastic approaches

Deterministic solutions (e.g., using traditional finite elements methods) which are nowadays very common in solving engineering problems require fully defined, certainly determined parameters and well-defined governing equations to result in useful and accurate outputs. Any input parameter must be known certainly (non-probabilistic) prior to proceeding the deterministic algorithms. In fact, the level of accuracy in the input will directly affect significantly the output with even much higher loss of accuracy due to accumulation of errors including approximation and computation errors. Furthermore, in a worse case (but common) if the phenomenon being studied is ill-defined (e.g., due to incomplete or inadequate information; Hsu 2008) then application of deterministic methods for the assessment, evaluation and modelling will be very limited. In practice, one has to choose even fewer parameters for models (i.e., further approximation) and often to assign values based on very limited observations in order to achieve reasonable computing cost. In either case, the result will be erroneous and thus inaccurate. For example, a system can be modelled as $X = f(\theta)$, where θ is set of parameters for adequately describing a phenomenon represented by X. Any failure in proper determination of parameters causes errors in X decreasing its reliability. Indeed, in real-world applications there is almost always a lack of sufficient data/information. In fact, if the phenomenon under study is complex and there is a significant lack of direct measurements (data) and associated information (which is particularly the case for fracture network modelling in hot dry rock systems located kilometres beneath the ground) a feasible solution would be that able to get utmost information from the available data of any sort. In fact, as an extreme case when there is no information on a subject, a guess is the only legitimate solution. For situations mentioned, stochastic methods (Chilès 2004) provide a means of quantifying the uncertainties (i.e., certainty requirement in traditional deterministic methods can be tolerated) arising from, inter alia, insufficient data, measurement errors and problem specification and of including these uncertainties in the modelling. Basically, stochastic methods use applied statistics and probability to tackle the issues in the data including lack of adequate data by addressing and incorporating associated

uncertainties. Random functions as the core requirement in most statistical methods play an important role in stochastic modelling, for example. In a nutshell, the power of stochastic methods comes from observation-wise (data and expertise based) use of properly defined random functions feeding to Monte Carlo methods in order to depict unseen aspects of data with the hope to increase the quality of processing (e.g., higher accuracy) and thus resulting in more accurate and reliable outputs (see further details in Cressie 1991; Jing 2003; Dowd et al. 2007). Stochastic methods have frequently been shown capable and efficient to model uncertainties involved in a variety of engineering modelling problems (CFCFF 1996) and their prospective future applications are rapidly increasing. Finally it is worth noting the recent trend in engineering analysis to combine deterministic methods with probabilistic outputs (stochastic models) in order to widen their applications and to improve their results.

1.2 Modelling

A mathematical model (Chernoutsan et al. 2011) is usually used to describe the most significant governing features of fracture networks (Dimov et al. 2011). If such an assessment is based on modelling of individual fractures the method is called *Discrete Fracture Network* modelling (DFN, Jing 2003; Jing and Stephansson 2007a) as opposed to continuum modelling in which the entire system is modelled as one continuous domain. DFN is more flexible (Dershowitz et al. 2000; Jing 2003) in dealing with the complex fracture configurations observed in practice using a stochastic solution where the distribution of fractures is often sparse and there is also significant uncertainty involved in measurement of fracture and fracture network parameters; thus deterministic methods such as finite techniques (e.g., finite element (FEM), discrete element (DEM), boundary element (BEM) and so on) in their traditional use (see previous section) cannot solely and satisfactorily handle those situations (see Jing 2003) although they are well-known and well-developed numerical modelling techniques and are applied to variety of engineering problems. Stochastic modelling may serve providing required input data for finite techniques by simulating realisations hence yielding a hybrid stochastic-deterministic approach. It is worth noting that such a

combination of deterministic and stochastic methods would have noticeable improvements in the resulting solutions, especially dealing with complex and less-known phenomena (a usual case in engineering problems).

1.3 Fractures and fracture networks

A fracture is defined as a discontinuity within a rock mass Priest (1993) and Jing (2003). The term thus also includes faults, joints, fissures, cleavages and even discontinuities between mineral particles (Lacazette 2011). By definition, a fracture can be on a scale of a few microns to several kilometres (e.g., faults). Engineering-scale fractures are generally greater than 10cm and less than 1km (Odling 1991). In its standard form a fracture is considered as an empty space between two parallel planes (Hernqvist 2009) embedded in a rock mass with the spacing between the two planes termed the aperture (Kacewicz 1994). A more complex representation can incorporate additional characteristics such as roughness of the fracture plane surface, the variation of the openings (Parisi et al. 2000) and even the tortuosity (Matyka et al. 2008) of the fracture geometry. The term *hydraulic aperture* (Koyama et al. 2009; Kvartsberg 2010) is used to describe the equivalent hydraulic behaviour of a fracture according to Darcy's model while taking into account all the deviations of the actual fracture from the idealistic parallel plane representation (Koyama et al. 2009). An ensemble of at least two fractures forms a *fracture network* (Figure **1.1**). In practice, even a simple fracture network can contain hundreds of fractures of different sizes and orientations. The geometrical intersections of fractures are extremely complex (Locsin and Einstein 2005) but for fluid flow through rock masses (Hayashi et al. 1999) fracture intersections are the critical control points for the behaviour of the system.

1.4 Rock mass

A rock mass is defined as an ensemble of rock materials (mineral particles) and all discontinuities within its entire volume (Figure **1.1**; see also Harrison and Hudson 2000). Rock materials are often called the *matrix*, in which discontinuities have a wider definition including joints, fractures, faults and pores (Jing 2003).

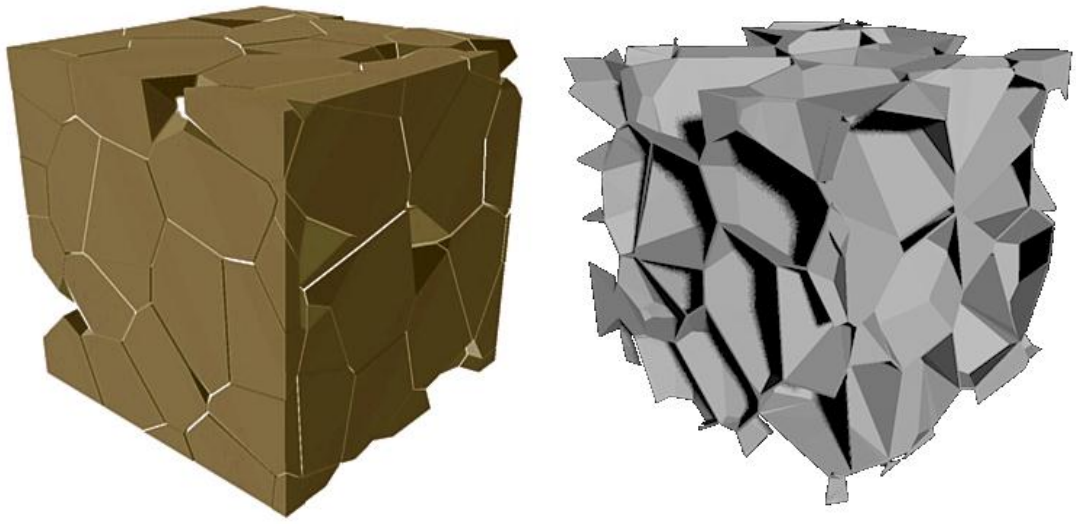


Figure 1.1: Rock mass. Rock blocks on the left which are of interest of rock mechanics and geotechnics engineering, and fractures (fracture network) on the right which are of interest of water resources, petroleum, geothermal and mining engineering.

With the above short discussions the title of this thesis makes now more sense in general, while in the following sections the research is introduced in details. Chapters (including papers) present comprehensive literature reviews, the details of topics in this research, their developments, discussions, proposals and applications. The ultimate goals of this research are summarised at the end of this chapter.

1.5 Summary of literature review

Fracture and fracture network modelling play important roles in a variety of engineering applications. The stochastic modelling of a fracture involves describing its geometry. A fracture network is consequently synthesised as an ensemble of generated fractures. A fracture network may exhibit spatial correlation. It is worth noting that in the recent decades rock fracture modelling has found numerous applications in a variety of engineering disciplines.

In mining and civil engineering, areas using rock fracture models include tunnelling, underground mining operations; rock dynamics; stability analysis (Grenon and Hadjigeorgiou 2003) of rock structures in surface and underground excavations (see review by Jing 2003 for details). In Water and Petroleum

Engineering, it has been reported that the flow of water in underground storages and aquifers shows significant inhomogeneity (Freeze 1975) mainly due to the sparsely positioned pathways, which are controlled by fractures and fracture network in rock and property variation of soil (see Kvartsberg 2010). In petroleum engineering, fracture models are used for the evaluation of oil-gas reservoirs; the prediction of the production rate (see Cacas et al. 2001; Nelson 2001).

1.5.1 Geothermal Energy System

Geothermal reservoirs can be classified to two broad types: hydrothermal and hot dry rock (HDR). Hydrothermal resources include sedimentary aquifers and non-sedimentary rocks such as naturally fractured volcanic rocks. The heat in hydrothermal resources is transported by means of water exists in the system. In HDR type, however, there is no fluid in the system; thus, external fluid is injected to transfer the heat. Enhanced geothermal systems (EGS) are deep (3 to 10km beneath the ground) HDR systems in which additional high pressure high volume fluid is injected to enlarge the existing fracture network and also to initiate new fractures. The fundamental factors in the productivity of geothermal systems especially EGS are heat, fractures and fluid flow (Hanano 2004; Singhal and Gupta 2010). These three factors define the scheme of heat drawdown from the geothermal system. Fractures and fracture networks in EGS play the key role in geothermal development by defining the *flow path* that *connects* injection and production wells of the reservoir (Hanano 2004), i.e., the connectivity of the reservoir. Furthermore, the permeability of the reservoir is determined by the hydraulic characteristics of the fractures forming the connection paths.

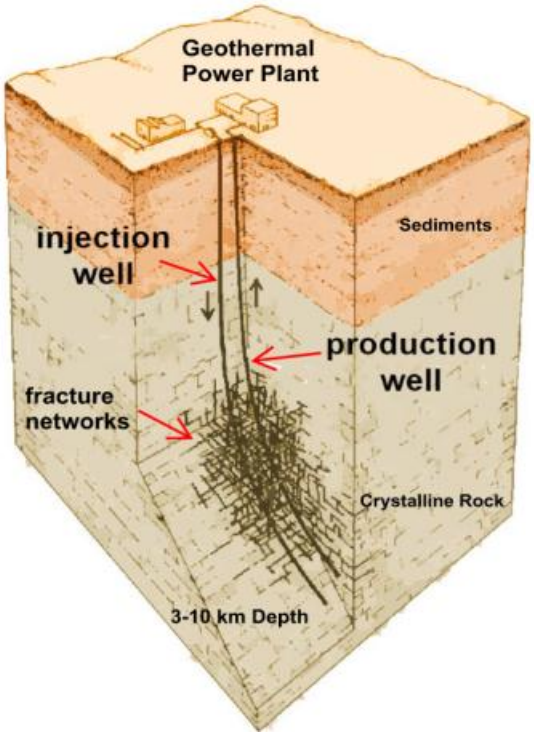


Figure 1.2: Enhanced Geothermal energy System (EGS). Cold water is pumped into the injection well to reach the geothermal heat. The contact between the fluid and hot dry rock is made due to existing and or stimulated fractures building an appropriate and efficient connected network for heat exchange (chamber). Redrawn and painted from MIT (2010).

In HDR EGS applications (Figure **1.2**), the permeability of the geothermal reservoir is enhanced by fracture stimulation as the natural fracture system in general has very low permeability. Studies have shown that the EGS host rock at depths of more than three kilometres is mainly granite. Granite is a hard crystalline rock that is largely impermeable. Hence fractures are the only pathways for fluid flow. On the surface, low-temperature fluid (currently water although super-critical CO_2 is being evaluated) is pumped down an injection well to the reservoir where it passes through the fractured rock absorbing heat and finally is extracted as steam from a production well. Fractures and fracture networks are critical in establishing the heat-exchange chamber, and in controlling the fluid flow and performance of the engineered reservoir (MIT 2010).

Rock fracture modelling is a multi-disciplinary problem involving mathematics, statistics and spatial analysis, petroleum, civil, geotechnical, and mining engineering, geology and geophysics, computer programming and data visualisation. The modelling of fracture networks starts by defining the geometrical

characteristics of the fractures followed by topological characterisation of the network. Any attribute associated with fracture geometry influences the response of the network to processes such as fluid flow. Topology determines directly the connectivity of the fracture network, which is crucial for fluid flow. Thus, it is important to make the geometric models as realistic as possible. The topology (here simply interconnections among fractures) of a fracture network is another important factor in the modelling. Note that although topology is a product of fracture assemblages, and thus cannot directly be modelled, it can be characterised effectively by means of fracture parameters, most importantly the locations, lengths and orientations.

1.5.2 Geometric modelling of fractures and fracture networks

Fractures and fracture networks in the real world are very complex in geometry and topology. Advanced techniques are thus required to model them at a scale appropriate for the evaluation of any process to which the network is subjected, e.g., fluid flow. Geometrical modelling of fractures can be associated directly and simultaneously with the topology characteristics of fracture networks. This may be done, for example, by pixel-based simulations (e.g., sequential geostatistical simulations, SGS; Deutsch and Journel 1998) or Voronoi tessellation simulations. In these methods, there is a strong dependency between fractures and the resulting topology (Jing and Stephansson 2007a). For example, in Voronoi tessellation, as fracture connectivity is a determining factor of network topology, once a Voronoi cell is built, all locally associated fractures are concurrently generated. Note that in this particular case, there will be no topology update on the local scale as the new fractures are added. A different method is to generate all fractures independently, first. Next the simulated fractures are located in space; that is, the associated topology is established then. The latter is known as Discrete Fracture Network modelling (DFN, Jing and Stephansson 2007b). It is widely used and well-developed. In DFN, using stochastic methods and point pattern analysis (e.g., point processes, Diggle 2003), the geometries, spatial characteristics, topological constraints, and other attributes can be incorporated effectively to produce realistic fracture networks.

Fractures are three-dimensional geometrical entities that are more or less complex both in the shape of each individual fracture (geometrical formation) and in the manner in which they are dispersed in space (spatial formation). The pattern of fracture locations, or fracture network dispersion pattern (Diggle 2003; Baddeley et al. 2006), is associated with the stress regimes in rock mass. The stress regime is formed due to physical, mechanical, chemical and other governing processes in the system so-called the underlying processes. Other attributes are orientation, size (~length), shape, aperture, roughness (see e.g., Dershowitz and Einstein 1988). These characteristics are basic geometrical and spatial characteristics describing the system of fractured rocks. They are subjects of modelling fractures and fracture network in order to achieve a realistic and so reliable picture of the system (e.g., geothermal energy system) which in turn helps to improve our understanding about governing processes. Therefore, the geometry of fracture and topology of fracture network are critical factors to describe the entire system in a sense that they define closely the behaviour of the fracture network against phenomena being studied such as fluid flow through fractures.

Two-dimensional representations

Fractures in two dimensions are represented by their trace lines on a sampling surface (a flat plane including outcrops, tunnel walls; example shown in Figure **1.3**), on borehole logging (core and images). The distinguishing characteristic of this type of modelling is that fractures (trace lines) are straight lines (often finite line segments). Therefore, the third dimension of a real three-dimensional fracture, which describes particularly the dip and the shape of fractures, is not incorporated in two-dimensional modelling. This type of modelling is carried out when there is no three-dimensional measurement available of fracture systems, as, for example, in sampling from outcrops in the field. Furthermore, they are shown useful for preliminary evaluation of concepts and proposals as a common practice in the literature.

For example, Baecher (1983) conducted statistical analysis of fracture traces to examine various biases involved in the field measurements. The work concludes that the orientation, size and censoring biases (Laslett 1982) are dominant. Chilès

(1988) investigated fractal and geostatistical properties of fracture networks. The work also proposed a framework to generate spatially correlated three-dimensional fracture networks using disk-shaped fractures. A review by Dershowitz and Einstein (1988) discussed various models for fracture networks in two and three dimensions mainly categorised into two groups: disaggregate and aggregate characterisations. In the former, statistical distributions (Figure **1.6**) are used for modelling attributes of each fracture, while in the latter the simulated fracture networks determine the attributes of resulting fractures. A more recent work by Renard and Allard (2011) reviews the lattice-based connectivity property of fracture networks mainly focusing on two-dimensional investigations. Interestingly, all the above example works are based on two-dimensional case studies. The main reason perhaps is that two-dimensional fracture networks provide a simple and useful framework for exploratory analysis. Exploratory analyses are better conducted on two-dimensional graphs due to their easier demonstrations. Note that in exploratory evaluations, visual inspection play critical role. Even though in some cases there is no further complexity in extending the method of analysis to three dimensions, e.g., in lattice-based connectivity analysis (see Pardo-Igúzquiza and Dowd 2003), two-dimensional studies still remain a primary choice for developing concepts and establishing theories; and it is also the chosen practice in this thesis. Often the only obstacle to extending a two-dimensional concept to three dimensions is the implementation of the associated computer code. Many of the ideas, developments and proposals in this thesis are based on two-dimensional fracture networks although in some cases three-dimensional extensions are developed, demonstrated and discussed, and some algorithms and/or pseudo-codes are presented.

Briefly, for two-dimensional fracture network modelling, as fractures are represented by line segments, the main focus is on determining location, length and orientation attributes. In a very simple case, fracture lengths are assumed to be infinite i.e., traversing the entire study area. Orientations can be limited to two orthogonal directions. Locations can be obtained from a uniform distribution function. In an advanced formulation, however, locations are determined by means of Poisson point processes (see Sects. 2.1.3 and 2.1.12 and Baddeley 2010)

providing plenty of patterns including homogeneous, inhomogeneous dispersion templates. Orientations follow a von-Mises distribution which is a bell-shaped distribution on the interval $-\pi$ to π. Lengths are obtained from power-law distribution functions including exponential and lognormal (Bour and Davy 1999; Sect. 2.1.12 and Figure **1.6**). The above procedure is sometimes referred to as a *Poisson line segment process* (Warburton 1980). Further details, demonstrations and discussions are given in Chap. 2.

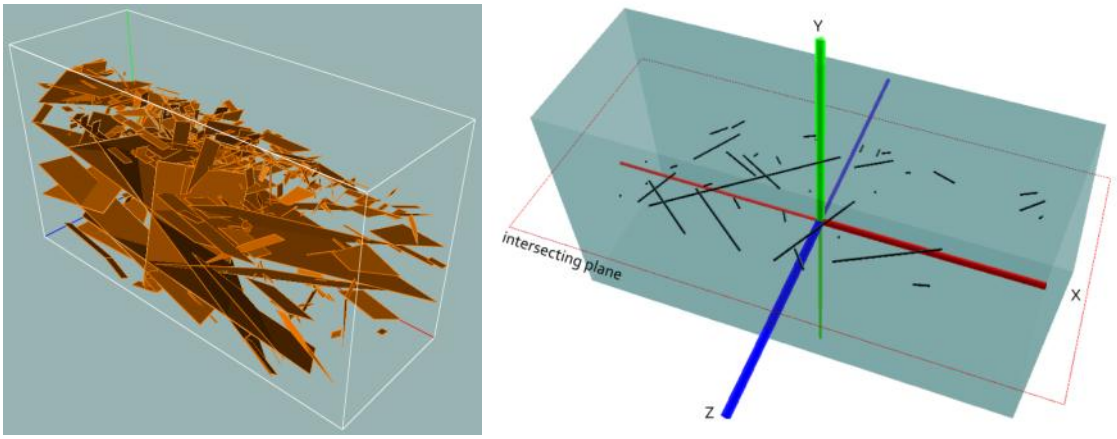


Figure 1.3: A real three-dimensional fracture network, the Leeds Rock Fracture Dataset (Dowd et al. 2009). On the right, fracture trace lines are generated by means of intersections between fractures and a horizontal plane (red rectangle). It follows that fracture trace lines can be simulated by means of appropriate distribution functions for their locations (e.g., Poisson), length (e.g., Power-law) and orientation (e.g., von-Mises).

Three-dimensional representations

In three dimensions it is common practice to model a fracture by a flat plane. In the simplest cases, it is modelled as an infinite plane, or as simple geometrical shapes such as a circle or an ellipse (Figure **1.4**); recommended reviews are Dershowitz and Einstein 1988; Staub et al. 2002. A more recent trend is, however, to model fracture networks as a set of polygonal fractures which are randomly positioned in space. Figure **1.3** shows real measurements of fractures in three dimensions. They clearly form polygons. Polygonal representations of fractures may serve as a flexible framework to address the complexity of real fracture networks (example shown in Figure **1.3**). Generally, a set of assumptions is made prior to building a model of fractures and fracture networks. For example, a fracture in modelling is assumed to have a planar shape (flat plane or ellipse etc.;

planar polygons). Note that each side of the fracture is called a *fracture surface*, which is commonly considered to be flat but can also incorporate roughness. The aperture spacing of a fracture is much smaller compared to the length of fracture i.e., five or more orders of magnitude smaller (Odling 1991). The length of a fracture is defined as the longest dimension of its shape. The most significant determining characteristics of a fracture in a three-dimensional representation include length (one or several scalars depending on the complexity of the fracture shape) and orientation (commonly two angles are adequate). A third important characteristic, the location of fractures completes the modelling stages for generating fracture networks. Hence, three-dimensional modelling follows the same procedure as two-dimensional modelling. Figure **1.4** presents chronological (almost from A to G) developments in the modelling of three-dimensional fracture networks. Only a brief explanation of these methods is given here. Three orthogonal orientations with infinite fracture sizes are the main features of the method (A; Snow 1965) in Figure **1.4**. It is simple but has no real application. A model proposed by Baecher (1983) in which fractures are circular disks is shown in Figure **1.4**B. In the *Enhanced Baecher* method (proposed by Geier et al. 1988) fracture disks can clip one another providing incomplete disks. The "*Baecher Algorithm Revised Terminations*" (BART) model (Dershowitz et al. 1998) in Figure **1.4**D is a generalisation of the Enhanced Baecher method by incorporating non-uniform random locations for fractures. Furthermore, in this method fractures are generated in two stages. The first generation acts as parent fractures around which the second generation is propagated (similar to the parent-daughter recursive method). The Veneziano model (E; Dershowitz 1984; Einstein 1993) in Figure **1.4** is significantly different from the others and in which fracture shapes are polygons resulting from intersection between pairs of randomly oriented Poisson planes (Dershowitz 1984). Model (F) in Figure **1.4** demonstrates the effect of incorporating the location point density resulting in fracture networks with varying density (see further in Dershowitz and Einstein 1988). Finally, model (G) in Figure **1.4** shows the use of a simple discrete fracture network modelling technique. Readers are referred to Staub et al. (2002) for a much longer list of methods with details. Regardless of the method of generating fracture network in Figure **1.4**, even wider patterns can be easily synthesised by advanced application

of marked point processes (Illian et al. 2008), which have been used in this thesis for two- and three-dimensional fracture networks. Also note that, most of the described models are too simplistic for satisfactorily modelling fluid flow for which the geometry and topology of fractures and fracture networks are determining factors. As a quick note, the use of arbitrary shaped polygons is recommended as can be seen in Figure **1.3**, for example, which is a real fracture network measured in a block of granite (Dowd et al. 2009).

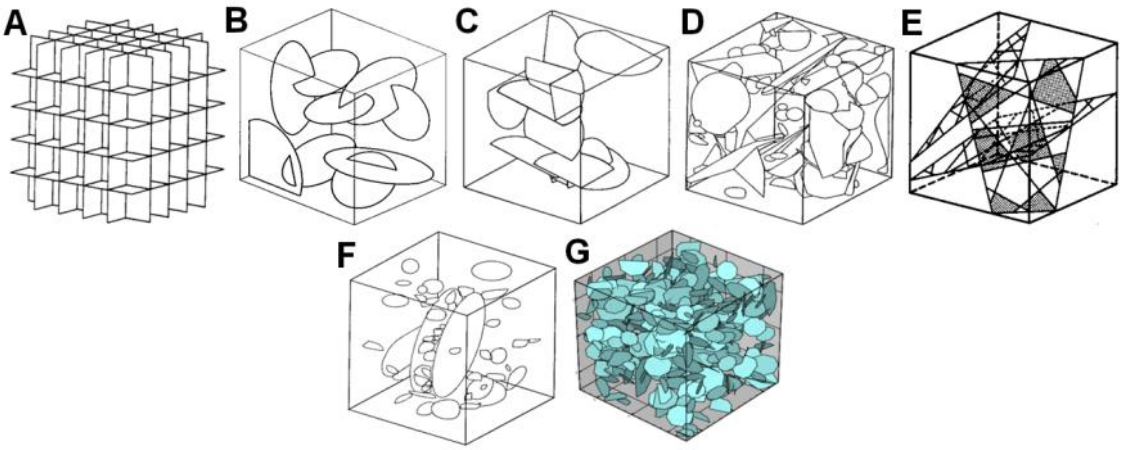


Figure 1.4: A historical visual review of fracture modelling proposals in the literature. (A) Orthogonal, (B) Baecher, (C) Enhanced Baecher (fractures can clip each other), (D) BART (random size), (E) Dershowitz (complex shapes on a plane), (F) Density model (inhomogeneous) and (G) Randomized polygonal shapes (images from Staub et al. 2002).

An example of a three-dimensional fracture network is shown in Figure **1.5** for which fractures are generated independently by means of distribution functions for lengths and orientations (details in the next chapter). The location of fractures in the network is determined by a Poisson point process. The number of fracture is 319 and the case is considered medium in complexity. This method overcomes all the difficulties in the previous models listed in Figure **1.4**. The termination procedure can be applied on fractures if required as a post-processing stage. One may use random functions (simple accept / rejection criterion) to determine which fracture is to be trimmed. Applying other types of constrains is also straightforward.

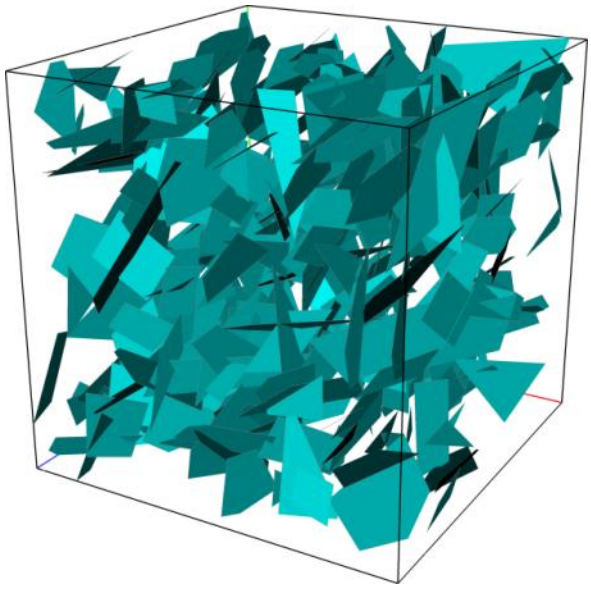


Figure 1.5: Realistic model of fractures and fracture network using polygonal shapes for fractures generated and distributed by means of marked point processes. Comparing this simulation with the real fracture data set shown in Figure **1.3**(left) shows a high match with reality.

Once a fracture network has been simulated and validated against the specified characteristics (e.g., distribution functions and reproducing conditioning data values where they exist) the next stage is to characterise the established model. This aim primarily is achieved by investigations on realisations from the models. The output of investigations may be generalised by means of E-Type methods for particular purpose. A number of examples are given in the following chapters.

Characterisation of models

Characterisation of a synthesised fracture network model depicts its significant features which are subject of comparison between different models. Features are density, statistical and spatial properties, intersections between fractures in the fracture network, connectivity (see also Chilès and de-Marsily 1993). Density estimations may include location points, line or plane density evaluations, intersection density and block density. Intersections between fractures build connected fracture groups (*clusters*). Spatial properties of fracture clusters and their statistics (e.g., connectivity measures) are also determined. Fracture clusters may provide pathways between two points (say, inlet and outlet stations in fluid flow modelling applications) which in turn provide a means to model flow of fluid

and or heat. Fracture network connectivity has been widely studied in the literature where it usually means cell-based (also called lattice- or grid-based) evaluation of connectivity (examples include Robinson 1983; Renshaw 1996; Bour and Davy 1997; Pardo-Igúzquiza and Dowd 2003; de-Dreuzy and Erhel 2003; reviews by Michaelides and Chappell 2009; and Renard and Allard 2011). While connectivity is seen between cells or in a grid, the focus of the reports is on cell (edge) neighbourhoods and is related entirely to percolation theory (Stauffer and Aharony 2003) in which any established connection between two sides of a study area is quantified. The large volume of work in percolation theory is based on grid networks providing two sub-systems: site percolation and bond percolation. For the use of percolation theory in its traditional form, fractures in two- or three-dimension need to be projected onto a grid (Odling 1991) or, discretised as cells. Both methods degrade the connectivity into the connectivity of grid lines or cells. Evaluation of connectivity in lattice-based methods is straightforward; however, induced biases due to degradation are not negligible. Whatever the method used to estimate the connectivity, the initial step in these methods is to discretise fracture networks into a course grid. Usually, the cell dimension in a grid compared to the fracture aperture is larger by several orders of magnitude. Note that the difference between the size of fracture length and its aperture is usually over five orders of magnitude i.e., the length is 10^5 times larger. Thus practically speaking, it is infeasible to increase the resolution of discretisation to decrease the effect of discretisation. As a result the *positive false* bias is particularly predominant. Chapter 6 of this thesis covers an extensive list of techniques and concepts for the characterisation of fracture networks.

Data collection and statistical inference of model parameters

Fractures can be sampled on surface outcrops (natural or man-made) in the field. This is common in geosciences for measuring joints, faults. Measurements may include the location, dip-direction (azimuth), dip (if possible) and the length of fracture traces. Data can also be gathered from underground works e.g., tunnels and mining activities. In other cases, drilling is used to access below the surface for data gathering (well-logging). There are several techniques for fracture sampling via boreholes. For example, measurement of the fracture traces on the cores or on

photographs taken of the walls of the borehole. Geophysical signals (electromagnetic, seismic etc.) can be used to penetrate deeper into the rock and measure the fractures around the borehole. For very deep cases such as EGS it is common practice to use the seismic events produced by injecting large volumes of high-pressure fluid into the rock mass to stimulate the propagation of fractures; these events are usually the only data available on any meaningful scale. Further development or holding of an active EGS is closely dependent to successful stimulations. Fractures under stimulation pressure are expanded. Some new but relatively smaller fractures are also created. During the expansion due to fracturing and hydraulic pressure movement of rock blocks occurs which generates micro-seismic events. Micro-seismic events are recorded by means of geophones. In Fadakar-A et al. (2013a) a practical characterisation of fracture networks by means of a specific random sampling technique (called *RANSAC*) on micro-seismic point cloud data has been proposed.

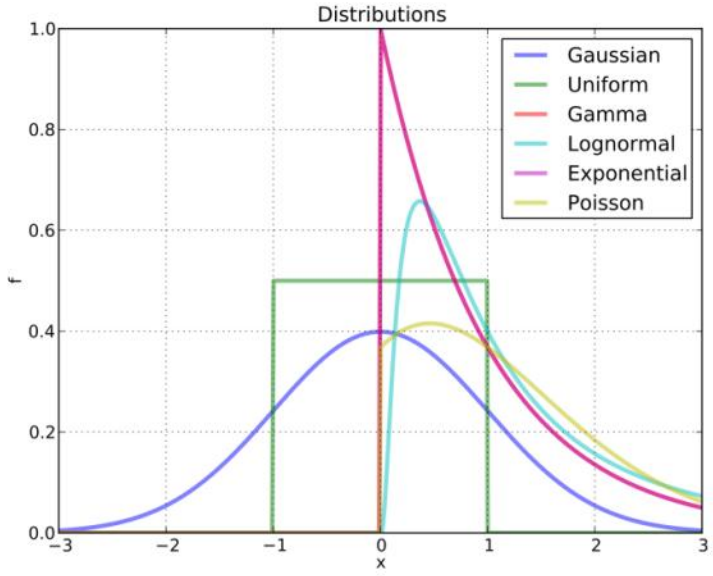


Figure 1.6: Commonly used distribution functions in fracture network modelling. Those without a negative tail and having a long positive tail are commonly used for modelling of the length of fractures. Gaussian distribution in its polar form i.e., von-Mises distribution is used for orientations. Poisson distribution is used indirectly to model the location of fractures in space satisfying complete spatial randomness criterion. This figure is not meant to exclude the use of any other distribution for any of the attributes of fractures and fracture networks.

1.6 Summary

Fracture and fracture network modelling is a truly multi-disciplinary research area. Although there is a large volume of publications is this area, the research is still challenging and very active. The complexity of realistic fracture networks in their geometrical characteristics, which in turn determine static and dynamic mechanical properties of rock blocks, lack of direct measurements, uncertainty involved in indirect measurements make the characterisation of fracture networks a very challenging problem. Having this said, the results of characterisation are essential in the evaluation of the response of fracture system against stress regime and fluid flow (Dverstorp 1991), for example. Thus as our understanding of the effective factors in the geometrical modelling of fractures and consequently topological properties of fracture networks increases, more accurate and thus more reliable results may be expected from corresponding analyses. As a determining factor the inter-connection between fractures needs to be well understood for evaluation of fracture network response against fluid flow, for example. To a limited extent, during simulation, the fracture network connectivity can be controlled by adjusting the defining distribution functions for fracture and fracture network parameters (locations, orientations etc.). Either a fracture network is a simulated model or measured in the field, a suitable means is required to characterise its connectivity properly. The loss of fracture connectivity information while discretising (which is predominant in traditional connectivity measures) is significant even if high resolutions (assuming them feasible) are used. This is basically due to fact that the aperture dimension of a fracture compared to its length is very small. A more accurate solution would be that if the discretisation is necessary the connectivity information between fractures to be kept invariant to the discretisation resolution. Such a method promises more reliable connectivity evaluations, indeed.

In Chap. 2, several issues in fracture and fracture network modelling are addressed resulting in practical solutions to simulate realistic fracture networks suitable for various needs and configurations. The proposals overcome the over-simplification issue found in the existing literature (see Figure **1.4**) regarding

geometrical modelling of fractures and the associated topological constraints. The algorithms are straightforward (thus practical), effective and capable of dealing with any type of complexity and pattern in the fracture network. In this chapter, terminologies are explained, demonstrations are given and related discussions are made. New techniques to generate fractures and fracture networks are also presented. It is critical to understand how fracture networks are generated in a way that keeping a high consistency with the observations in the field if exist. Chapter 3 is a conference paper with the main focus on fracture network modelling in which some new findings are presented. Next, Chap. 4 is a journal paper on generating fracture networks conditional to real micro-seismic events data. Chapter 5 is another journal paper on modelling fracture networks incorporating spatial characteristics. Next, Chap. 6 is a comprehensive set of techniques and novel contributions to characterise fracture networks from various aspects. This chapter is dedicated to characterising fracture networks with the intention of providing practical tools for detecting hidden features of fracture networks. This helps to establish criteria for comparing fracture networks, associating their behaviour with the model characteristics and, ultimately, increasing our knowledge of fracture networks. Chapter 7 is another journal paper on a new proposal, connectivity field, to characterise the connectivity of fracture networks without loss of inter-connection information of fractures, as is the case with traditional lattice-based connectivity methods. Applications of connectivity measures, particularly the connectivity field and connectivity index are presented in Chap. 8 which is another journal paper and also a conference paper. Chapter 9 is another conference paper on further applications of connectivity measure to optimising well locations in fractured-based reservoirs. The thesis is concluded with Chap. 10.

List of Papers (C: Conference; J: Journal):

- A General Framework for Fracture Intersection Analysis: Algorithms and Practical Applications. (C, Chap. 3)
- The RANSAC Method for Generating Fracture Networks from Micro-Seismic Event Data. (J, Chap. 4)
- A Spatial Clustering Approach for Stochastic Fracture Network Modelling. (J, Chap. 5).

- Connectivity Field: A Measure for Characterising Fracture Networks. (J, Chap. 7)
- Connectivity Index and Connectivity Field towards Fluid Flow in Fracture-Based Geothermal Reservoirs. (C, Chap. 8)
- A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs. (C, Chap. 9)

1.7 Research objectives

In this research the main focus was on characterising fracture networks with emphasis on evaluating and improving existing methods, proposing and developing new concepts and methods, and an attempt to bridge the gaps between theoretical and real world applications. The research has followed a step-by-step development by firstly working on theoretical concepts (Chap. 1 and 2), then on fracture network generation frameworks (Chap. 2, 3, 4 and 5), on methods to characterise generated fracture networks (Chap. 6, 7, 8 and 9) and finally on connections between the proposed or discussed theories, introduced methods and real world applications with case studies (Chap. 8 and 9). Briefly, (*i*) fracture network modelling, (*ii*) characterising fracture networks and (*iii*) applications of the research findings are three main components (and stages) of this research/thesis. Each stage includes fairly a comprehensive study of fundamentals, issues, existing solutions and introductions to the original proposals and solutions presented in this thesis.

Within the context of the above objectives there are several secondary objectives (computer codes, EGS applications/case studies and so on), processes to generate DFN based on original concepts of fracture development, extension and termination; and theories to facilitate the visualisation, quantification and interpretation of network characteristics have also been developed. A complete Matlab package was also programmed (Appendix 2) to facilitate the development of the ideas, theories and the evaluation of the results. In the application of DFNs to geothermal reservoirs, the development and then application of several theories on lengths, apertures and connectivity of fractures and how these characteristics influence the flow of fluids through networks are presented.

1.8 Fracture network modelling

In the next chapter as the first stage of the triple study stages (mentioned in the previous section) in this research, fracture network modelling is introduced, existing methods are reviewed and several original concepts, theories and methods are proposed and discussed. As the understanding of fracture network generation is improved better understanding can be achieved in the subsequent stages i.e., characterisation and applications. All these contribute to clearer understanding of various aspects of fracture networks both in theory and real world applications.

2

Fracture Network Modelling

In the previous chapter a brief literature review of the fundamental understanding of the concepts being studied in the research was presented. As the first main stage of this research (also a main component of the thesis) fracture network modelling is a critical stage as it affects significantly all the subsequent stages. A more realistic fracture network model promises more accurate and more reliable characterising and application stages. A fundamental study of fracture network modelling concepts and methods is presented in the following sections. This chapter may also form a useful future reference on the concept of fracture network modelling.

2.1 Definitions and terminology

2.1.1 Point

A point is the simplest geometrical object with zero dimension but fully specified by its *location* in $\mathbf{E^d}$, where $\mathbf{E}$ is the Euclidian space in $\boldsymbol{d}$ dimensions (Corrochano 2005). A fracture as a three-dimensional object can be represented by a point (e.g., its geometric centre, see next section) thus its location is simply described as coordinates $(x, y, z) \in \mathbf{E^3}$. Similarly, fracture clusters (Sect. 6.13.1), fracture hyper-clusters and fracture networks can also be located by a point.

$$\mathbf{P} = \{(x_1, x_2, x_3, \dots, x_d) | x_1, x_2, x_3, \dots, x_d \in \mathbf{R}\}, \qquad \mathbf{P} \in \mathbf{E^d} \tag{2.1}$$

From the definition in (1), the *location* of a point can have arbitrary number of dimensions ($\boldsymbol{d}$); while commonly a pair of two (i.e., $\mathbf{P} = (x, y)$ for fractures in two-dimensional space, fracture traces) and three (i.e., $\mathbf{P} = (x, y, z)$ for fractures in three-dimensional space) real numeric values is used (Haining 2004). These examples refer to usual meaning of the term dimension which is based on spatial distances. Adding the time as the fourth dimension which is already common (e.g., in seismic investigations) advances the definition of location into a combined measure of spatial and temporal differences. In a more advanced and slightly abstract application, the term *location* can be extended even further to incorporate any additional attributes such as proximity to neighbouring points and density of points (Illian et al. 2008). This type of representative feature abstraction has many

other applications and, for points, it has been developed extensively as the theory of marked point processes, for example. Objects can be represented by points irrespective of their size, shape, complexity and frequency. Attributes can be associated then with the points as marks. This method of representation is used throughout this thesis for evaluating and developing ideas and proposals.

2.1.2 The Centre of Geometry

The most representative location point for a geometrical object (**O**) is its Centre of Geometry (**CoG**, or geometrical centre; Murayama and Thapa 2011). For geometrically complex shapes **CoG** is preferred over the commonly used and physically intuitive centre of mass (**CoM**, Weltner et al. 2009). The difference between these two centres is better apparent when there is a spatially inhomogeneous dispersion of shape-determining points on the boundary of object. For example, assuming three-dimensional fractures are flat polygon objects with either evenly distributed mass or fully void the **CoM** can easily be found by averaging the coordinates of their vertices (i.e., boundary points). However for fractures with spatially-uneven located vertices the **CoM** will be closer to the denser areas in terms of number of points thus will be biased (Figure **2.1**). The **CoG** incorporates the distance between two adjacent vertices, i.e., edges. Hence every fracture, whether two- or three-dimensional, is an object that can be robustly located by its **CoG** (see Figure **2.1**).

$$\mathbf{CoG} \sim \mathbf{O} \rightarrow \mathbf{P} \tag{2.2}$$

As a result of representing fractures by their **CoG** (a single point) they can be analysed by several well-developed methods including Point Processes (Diggle 2003; Baddeley et al. 2006; Baddeley 2010) and Spatial Analysis (Murayama and Thapa 2011). Some applications are discussed in Chap. 6.

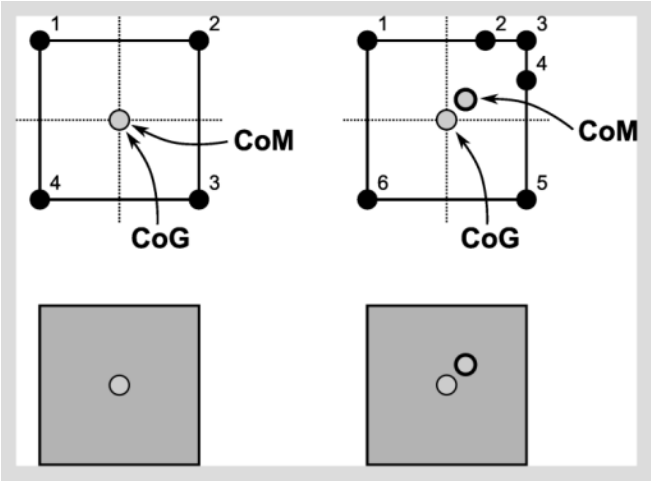


Figure 2.1: CoG vs. CoM; CoG is resistant against density of points (vertices) and thus more suitable for representing fractures.

2.1.3 Point processes

In a broad sense, point processes are *stochastic models* applied to *point patterns* (Illian et al. 2008). Point patterns, either regular or irregular, can be generated by means of stochastic simulations. A particular regular pattern can also be considered as a customised realisation of a stochastic model. A realisation of a point process is often called a *point pattern* (Diggle 2003). Pattern analysis (Corrochano 2005) is a broad branch of computer science and is applied to points and other objects. An object, regardless of the complexity of its shape or any other attribute, can effectively be modelled by means of point processes, associating each attribute as a mark (Descombes and Zerubia 2002) of the point process. The effectiveness of application of marked point processes in fracture network modelling is basically affected by the level of representativeness that the model offers. Every mark is modelled in order to address some important attributes of the fractures or fracture networks. There is a trade-off, however, between the number of marks and the complexity of model for which an ultimate solution would be case-based evaluations.

2.1.4 Line

A line is the simplest one-dimensional object and is completely specified by its end-points, $\mathbf{P_1}$ and $\mathbf{P_2}$ (Weltner et al. 2009). It can also be specified by a point and an angle (α). Note that the term line usually means a finite line i.e., a segment. Thus, for a complete description of the second form the length (l) is required.

$$\mathbf{L} = \{\mathbf{P}_1, \mathbf{P}_2\} \quad \text{or} \quad \mathbf{L} = \{\mathbf{P}_1, \alpha, l\} \tag{2.3}$$

The intersection of a fracture with an exposed surface, such as an outcrop, appears on that surface as a line, commonly called a fracture trace line (Odling 1992) or simply a *trace*. Without loss of generality one may consider a trace as a straight line. The intersection of two traces is therefore a point. In three-dimensions the intersection of two fractures is a *line* or a *point* depending on the shapes of the fractures and their relative arrangement in the three-dimensional space (position and rotation; refer to Chap. 3 and or Fadakar-A et al. 2011).

2.1.5 Polygon

A set of connected lines forms a polygon. Fractures in three dimensions can be modelled by convex polygons (Goodman and O'Rourke 2004). Curved fractures can be easily flattened onto some tangents producing a set of simple convex flat polygons. Hereafter polygon is meant as flat polygon. Convex polygons are standard for robust geometrical operations (e.g., intersections; Toussaint 1985). A quadrangle, quadrilateral which is the simplest four-vertex convex polygon, is commonly used to model three-dimensional fractures (Blocher et al. 2010). Any quadrilateral representation of a fracture can be approximated by an ellipse. The elliptical representation of fractures is commonly used, and widely reported in the literature (CFCFF 1996), primarily because of its simplicity in parameterisation. It is worth noting that there are, however, significant errors in this simplification especially in the approximation of fracture intersections. In light of CoG concept (see Sect. 2.1.2), it can be generalised that there are very rare cases in which CoG of an ellipse would match CoG of the original polygonal fracture. Also note that the area and permitter of a modelled ellipse would be significantly different than of the

fracture. These errors are more problematic when dealing with fracture network modelled by ellipses in order to analysis the connectivity, intersection and topology. In addition, any statistic inferred from elliptical approximations of fractures is likely heavily biased. On the other hand, polygonal modelling avoids all the above mentioned problems as it matches any complexity in the shape of fractures either perfectly or with a negligible approximation. Polygonal representation delivers: high effectiveness in modelling, high performance computation, high accuracy in fitting, high flexibility in dealing with complexity in fracture shape and high realism in modelling of fractures and fracture networks.

2.1.6 Convex-hull

For any subset $\boldsymbol{S}$ of points in $\mathbf{E}^{\mathbf{d}}$, the convex-hull is the smallest convex set containing $\boldsymbol{S}$ (Preparata and Shamos 1985). A convex-hull is also a *polytope* defined by its boundary. In two dimensions, a convex-hull of a set of points is a convex polygon and in three dimensions it is convex polyhedron. Fracture network in general while is represented by points either **CoG** or *vertices* can be covered by convex-hulls. Several algorithms can be used to compute convex-hulls for points in two and three dimensions, including: Graham's scan, Jarvis's march, quick hull, divide-and-conquer, dynamic convex-hull for points in two dimensions; gift-wrapping and beneath-beyond for points in three dimensions (see Preparata and Shamos 1985 for details).

2.1.7 Smallest enclosing circle, ellipse, sphere and ellipsoid

The smallest enclosing circle (**SEC**), ellipse (**SEE**), sphere (**SES**) and ellipsoid (**SEEd**) are the smallest circle, ellipse, sphere and ellipsoid, respectively, enclosing a set of points (Gartner and Schonherr 1998). If the shape of a fracture in three-dimension is approximated by a circle or an ellipse, **SEC** and **SEE** are useful to determine the size and orientation parameters, for example.

2.1.8 Largest Empty Circle, Ellipse, Sphere and Ellipsoid

An empty part of a space can be generalised into a largest empty circle (**LEC**), ellipse (**LEE**), sphere (**LES**) or ellipsoid (**LEEd**) (Toussaint 1983). For fracture

networks determining the size of an empty space can be useful for a variety of purposes including, but not limited to, safety risk evaluation and reservoir expansion (e.g., in enhanced geothermal systems). This is an important task for locating stations of hazardous waste storages beneath the ground in proximity of fractured areas. It has also applications for determining sub-areas with minimum probable tectonic activity. These measures can also be useful for characterising fracture networks to predict the productivity of reservoirs in terms of potential flow. Refer to Sect. 6.5 for demonstrations.

2.1.9 Triangulation

Triangulation (Boissonnat and Yvinec 1998; Okabe et al. 2000) is a procedure to fit a set of non-overlapping triangles to a finite number of points. For points in three dimensions the computational complexity of triangulation is $O(n \log n)$. It is used to generate two-dimensional *simplexes* (i.e., triangles) from sampling points in a study region or on a fracture surface. In three dimensions a fracture can be efficiently triangulated using any of the available algorithms (refer to Sadoyan et al. 2006). The resulting triangulation gives better performance and robustness for geometrical operations and is also a standard for visualisation. In addition, triangulation is vital part of meshing (Persson 1997) in any finite differential method in engineering including finite elements methods (Liu and Quek 2003; Paluszny and Zimmerman 2011).

2.1.10 Delaunay and Voronoi tessellations

The main purposes of triangulation are to model surfaces and to estimate proximity. The resulting triangles can be used to create the Delaunay (Boissonnat and Yvinec 1998) and Voronoi tessellations (Boissonnat and Yvinec 1998) which are standards for partitioning the space. Delaunay and Voronoi tessellations are dual and so each can be constructed by means of the other. In two-dimensions a Voronoi tessellation is a convex polygonal partitioning of a space while in three-dimensions it is set of convex polyhedrons. A Voronoi polygon is defined as:

$$\mathbf{V}(i) = \bigcap_{i \neq j} \mathbf{H}(\mathbf{P}_i, \mathbf{P}_j) \tag{2.4}$$

where $\mathbf{H}$ is the half-plane containing the set of points closer to $\mathbf{P}_i$ than $\mathbf{P}_j$. An example of Voronoi tessellation (sometimes also called a *Voronoi diagram*, *Voronoi partition* or *Voronoi decomposition*) is shown in Figure **2.2**. Note the treatment of the edge effect in Voronoi diagrams. Figure **2.2**(left) shows a Voronoi diagram in which no edge correction is applied. Two common approaches to edge correction are the periodic network (Okabe et al. 2000) and the buffer zone, each of which use additional points surrounding the region to be tessellated. In the periodic solution every edge meets the opposite edge whereas in the buffer zone correction the additional points are generated using random functions. Moreover, in the periodic solution the additional points are exact, translated copies of the original points and there is thus an implicit assumption of stationarity and homogeneity in the point pattern. If the pattern is not homogeneous, (e.g., with a systematic trend in point density from left to right) the periodic approach gives even worse results (Figure **2.3**). In the buffer zone method, the additional points along each edge of the region can be chosen such that to reflect the local patterns of points near those edges.

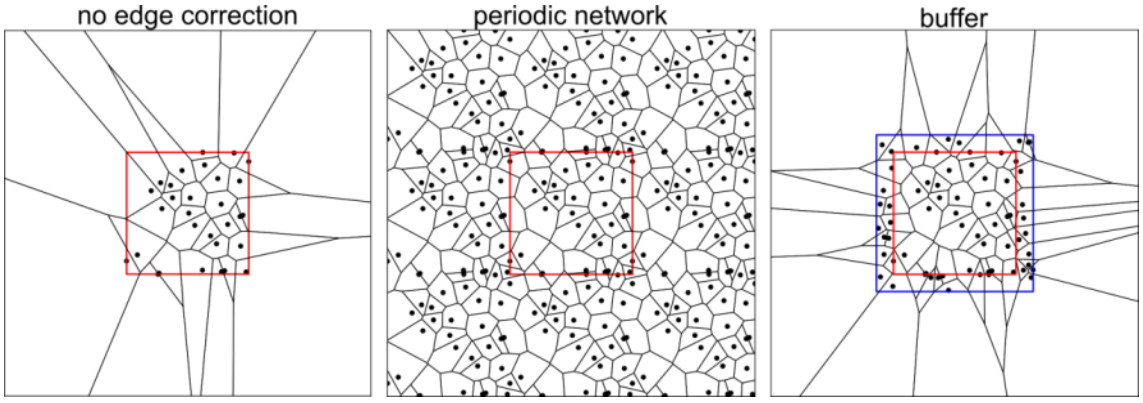


Figure 2.2: Edge effect for a Voronoi diagram. (a) no correction, (b) periodic network correction, (c) buffer zone solution.

It can be concluded that there is no uniquely optimal choice of edge correction for a given point pattern (see Figure **2.3** for an example of issues). Eventually, depending on the purpose of the application and the pattern of points, the choice of

method is determined, somewhat subjectively, by the extent to which a satisfactory tessellation is achieved. As shown in Figure **2.3** visual inspection may be required to avoid additional complexity and biases. For the example shown, if the purpose of tessellation is to partition the study region according to the locations of sample points, a noticeable number of Voronoi cells are excluded (red blocks) and thus, in this case, the periodic network edge correction is not recommended. Finally, it should be noted that if no edge correction is made to a Voronoi tessellation it is basically assuming there are no points outside in the neighbourhood of the study region.

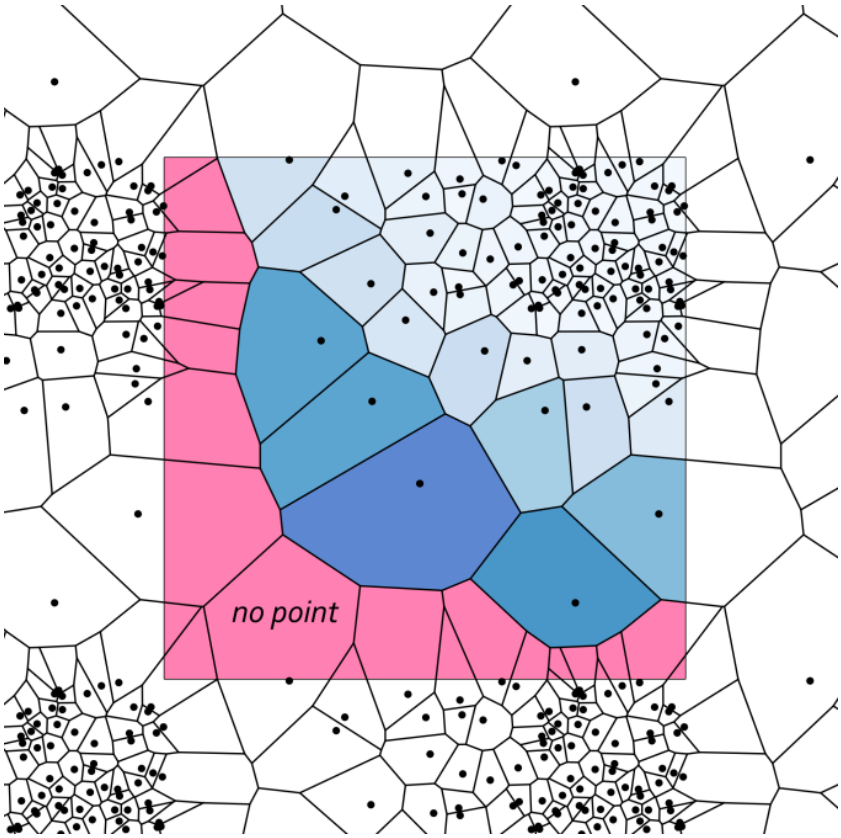


Figure 2.3: The periodic boundary edge correction of a Voronoi diagram may fail to generate a satisfactory tessellation if the point pattern is not homogeneous. The red parts are Voronoi blocks (cells) that do not contain a point and these are, thus, redundant. Shortcomings such as this inhibit the application of the periodic method as the associated error is apparently greater than not applying any edge correction.

2.1.11 Mathematical Morphology

The theory of mathematical morphology (Serra 1983; Kimmel 2003; Illian et al. 2008) stemmed from set theory to characterise and transform geometrical

structures (shapes). It has also been widely used in computer sciences particularly for image processing (Serra 1983). Some demonstrations can be found in Kimmel (2003), for example. Two fundamental operations are dilation δ and erosion ϵ that are denoted as $\oplus$ and $\ominus$, respectively. They are defined as follows (Kimmel 2003).

$$\delta_B(A) \equiv A \oplus B \equiv \bigcup_{b \in B} \{(A + b)\} = \{a + b : a \in A, b \in B\} \tag{2.5}$$

and

$$\begin{aligned} \epsilon_B(A) &\equiv A \ominus B \\ &\equiv \bigcap_{b \in B} \bigcup_{a \in A} \{a - b\} \\ &= \{x : x + b \in A, \forall b \in B\} \\ &= \{x : x + B \subset A\} \\ &= \bigcap \{A - a : b \in B\}. \end{aligned} \tag{2.6}$$

Two other basic operations are closing (●) and opening (○) and are defined as follows (see Figure **2.4** for demonstration).

$$A \bullet B = \delta_B\big(\epsilon_B(A)\big) = (A \ominus B) \oplus B, \tag{2.7}$$

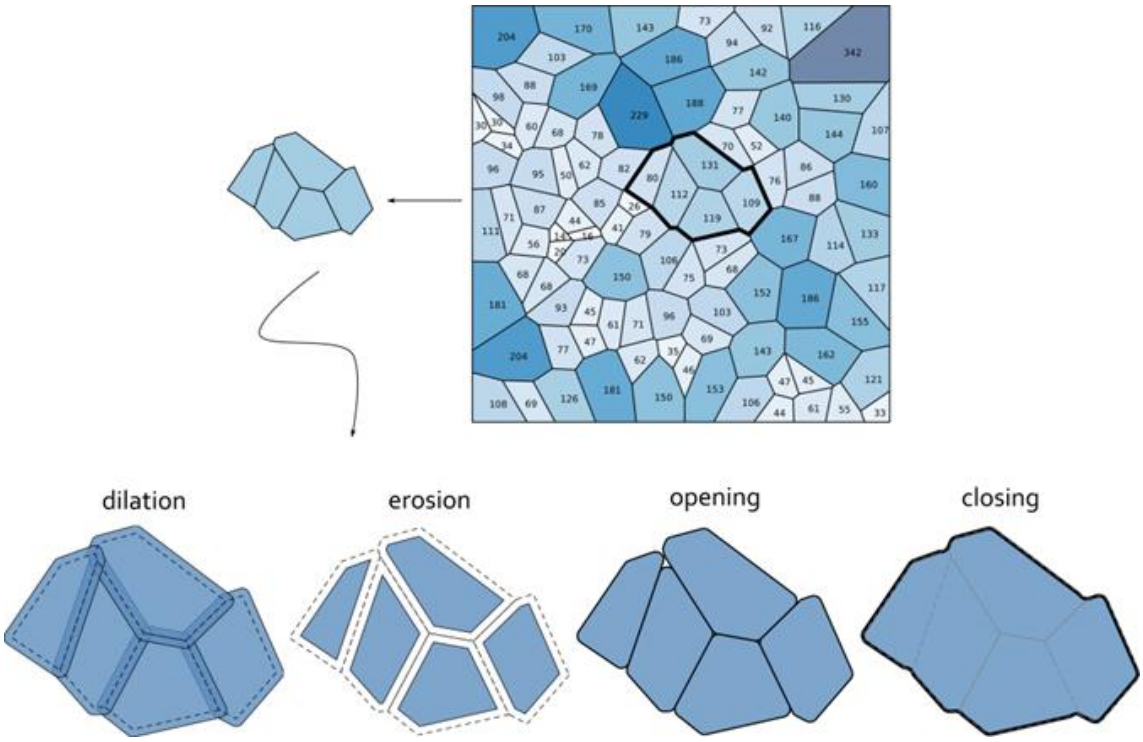


Figure 2.4: Fundamental morphological operations. Opening is combination of erosion and dilation; while in closing first dilation then erosion applies. Numbers in the main map are areas of blocks. Note that opening operation is more like mechanical erosion i.e., narrower the block more chance to be completely dismissed. Also note that how closing fills opened areas.

If the medium consists of two distinguishably separable types, say, the fractures and the matrix in rock mass, a binary map of it can be subject of opening and closing operations. Commonly used reservoir simulations include indicated (binary) maps of few facies (also called *Boolean* maps, see Vogel 2002), which are suitable for ordinary mathematical morphology operations. The aim may be decreasing the noise by removing small isolated parts, for example, which can be easily achieved by application of opening and closing in an iterative manner.

Synthesising fracture networks

Three methods are used in this thesis for modelling and generating fracture networks. They provide a reasonable coverage of the forms (i.e., *patterns*) of fracture networks as observed in practice. The first method is *Marked Point Processes* (MPP, see for details: Diggle 2003; Illian et al. 2008; Baddeley 2010) in which fracture attributes are stochastically defined. Stochastic methods are

powerful to deal with lack of data (information). They provide means to encapsulate the uncertainty associated with the input data and also the resulting information. They are fundamentally based on statistical inference and so may provide more flexibility in the determination of governing factors. A point process is used to simulate fracture locations and marks refer to the attributes of the fractures (e.g., size, orientation) which are simulated at those locations. Marked point processes do not explicitly simulate interactions among fractures, such as intersections, terminations of one fracture by another or propagation of fractures (approaches to these are discussed in Sect. 2.4). The second, and most commonly used, method of stochastic fracture modelling is *Discrete Fracture Network modelling* (**DFN**, Dershowitz et al. 2000; Jing 2003; Jing and Stephansson 2007b) in which each fracture is generated independently of all others. DFN can provide very useful fracture models for analysing the flow of fluid through fractures, for example. The connectivity of fracture networks can be efficiently evaluated from the generated fracture networks (Xu et al. 2006; Renard and Allard 2011; Fadakar-A et al. 2012; Fadakar-A et al. 2014). More sophisticated methods are required when there is a need to model blocks of rock (*Block Theory*, Goodman and Shi 1985; Windsor and Thompson 1996; Staub et al. 2002) produced by intersecting fractures (some solutions are discussed in Sects. 2.1.15, 2.1.16, 2.1.17 and 2.4).

2.1.12 Discrete Fracture Networks

DFN starts by generating the location of each fracture in the study region. This is done by generating location points (e.g., representative points of fractures, say, CoG). In the next step, the size of fractures (*length* in two-dimensional cases) is generated and an orientation is assigned to each fracture. These three attributes i.e., location, size and orientation are essential marks; additional shape attributes, such as aperture, can be incorporated as complementary marks. In addition to these geometrical attributes there are other physical (mechanical) properties, such as roughness, porosity, permeability and transmissivity (CFCFF 1996; Singhal and Gupta 2010) that are important to be modelled properly. The concept of marks may assist to build a proper model for these attributes. Conceptually, in the DFN methodology there is no limit to the number of attributes to be incorporated in modelling.

Locations

The point location of a fracture is the most fundamental attribute in DFN and it is initially defined by means of distribution functions (Kendall 2003). The resulting point pattern can be adapted more by applying concepts of point processes (Illian et al. 2008) such as thinning. The simplest pattern for a set of points is a regularly spaced grid (see Figure **2.5**a), i.e., the systematic sampling pattern (Haining 2004). A more usual point pattern can be constructed by means of a uniform random function (Figure **2.5**b) in which points are independently distributed in the study region. An alternative to uniform randomness is the clustering scheme in which points are distributed unevenly over the study region (Figure **2.5**c). These three forms cover most patterns as any pattern other than these three can be made up by combinations of the three forms. If a pattern is stationary (invariant) in space, it is called *homogenous* otherwise *inhomogeneous* (see example in Figure **2.6**, Diggle 2003; Baddeley et al. 2006; Illian et al. 2008). An inhomogeneous pattern is a point process that has a variable density depending on the location in the study region. The density function (Xu et al. 2003a) can be simple or complex, linear or higher order. For example, a multi-Gaussian density map (parametric estimations, Xu et al. 2003b) can be used to generate clustered point patterns (see examples in Figure **2.5** and Figure **2.6**).

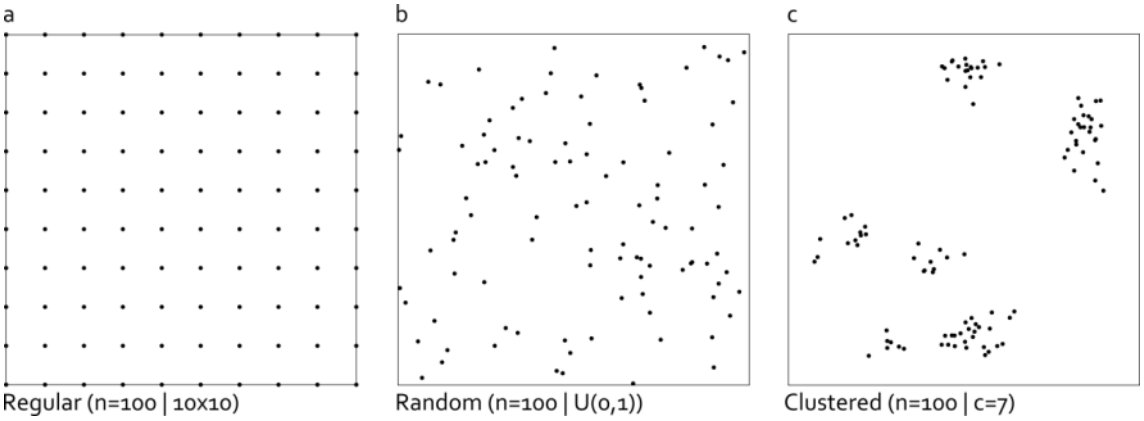


Figure 2.5: Point patterns: (a) regularly spaced, (b) randomly located and (c) clustered. Note that in all three patterns the number of points is 100; however, the resulting forms are significantly different.

Random patterns are simulated using any random functions including uniform, Gaussian, Poisson, exponential distribution functions. Of these, the homogeneous

Poisson pattern is statistically known as a standard example point process which satisfies the *Complete Spatial Randomness* (CSR, Diggle 2003) criterion. A side comment is that depending on the subject of study other types of distribution functions can be justified with no limitation.

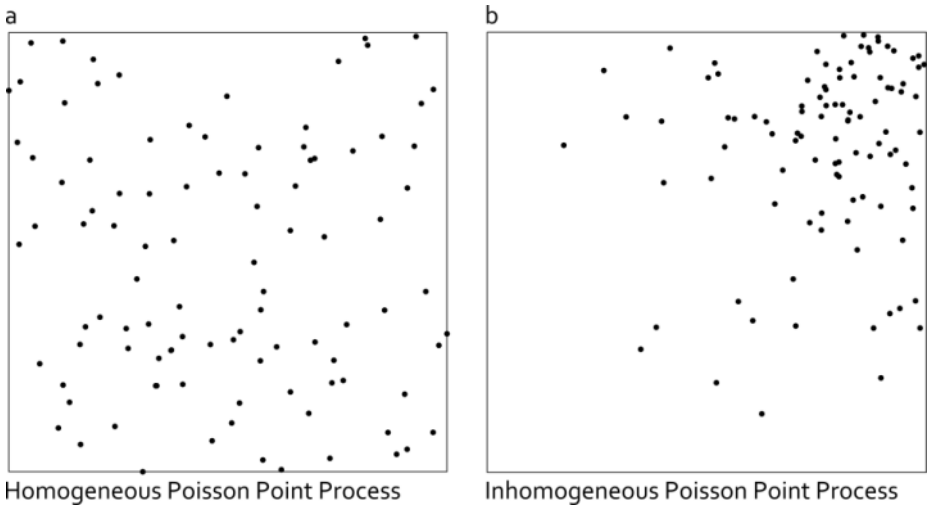


Figure 2.6: Point patterns can be homogeneous or inhomogeneous. Inhomogeneous point patterns expose intensity function rather than a single density value for the points. In (b) the intensity function is $\Lambda = x^{\alpha y}$ in which $\alpha = 0.83$. Both (a) and (b) have 100 points.

In a similar way, location points can be generated for three-dimensional fracture networks as shown in Figure **2.7** (see further discussion in: Baddeley et al. 2006 and Illian et al. 2008).

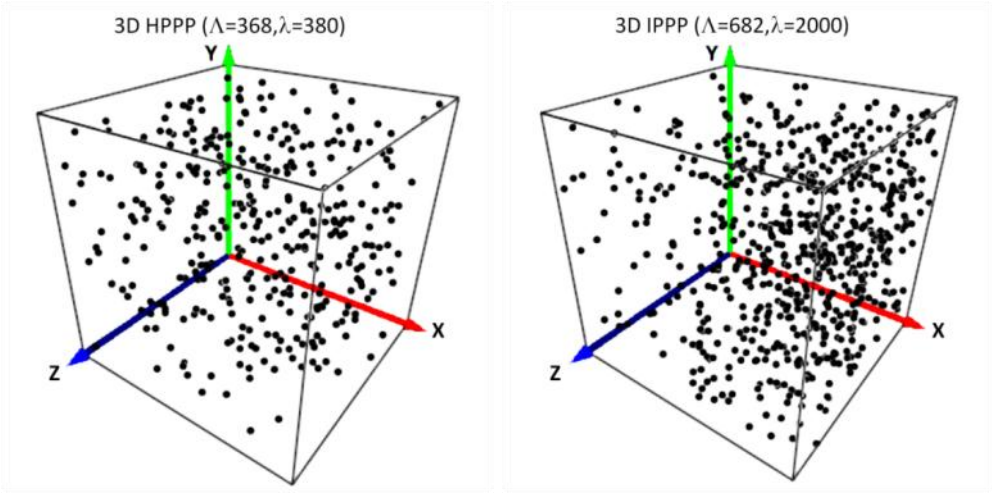


Figure 2.7: Realisations of points in three-dimensions using (left) homogenous Poisson point process with the resulting intensity of 368 and (right) inhomogeneous Poisson point process (IPPP) with the resulting intensity of 682. The intensity function for IPPP was $f(x, y, z) = 10x^2 + y^2 + z^2$.

A common approach to generating inhomogeneous point processes (Diggle 2003; Xu and Dowd 2010) is to generate a homogeneous Poisson point process with a global intensity function and then to thin the process by applying retention probabilities. The intensity function may be a well-defined parametric function in the space, a fine grid of individual density values, for example. A point is retained if either its local estimated density (Xu et al. 2003a; Freeden et al. 2010) is less than the corresponding grid cell or it fails the Metropolis rejection criterion (Kendall 2003; Illian et al. 2008).

Orientations

The orientations of fractures in rock masses are determined by the mechanical, stress field (Cosgrove 1998), geochemical and other factors, on local and regional scales. Probability distributions are convenient and widely used ways of summarising fracture orientations. The most common distributions used for fracture orientations in rock masses are the von-Mises (or the circular Normal) for two dimensions, the Fisher (or von-Mises-Fisher) distribution (a von-Mises distribution on the $(n-1)$-dimensional sphere in R^n) for three dimensions and the uniform distribution (Dershowitz and Einstein 1988; Odling 1992; CFCFF 1996; Gringarten 1997; Fouche and Diebolt 2004; Gupta and Adler 2006; Fadakar-A et al. 2011). Some typical examples are shown in Figure **2.8**: oriented, semi-oriented and with no particular orientation. Although there are significant visual differences, all three sets of orientations, were generated from von-Mises distributions by setting κ equal to 1000, 10 and 0 for sub-figures a, b and c, respectively. The mean orientation is zero (horizontal) for all three simulations.

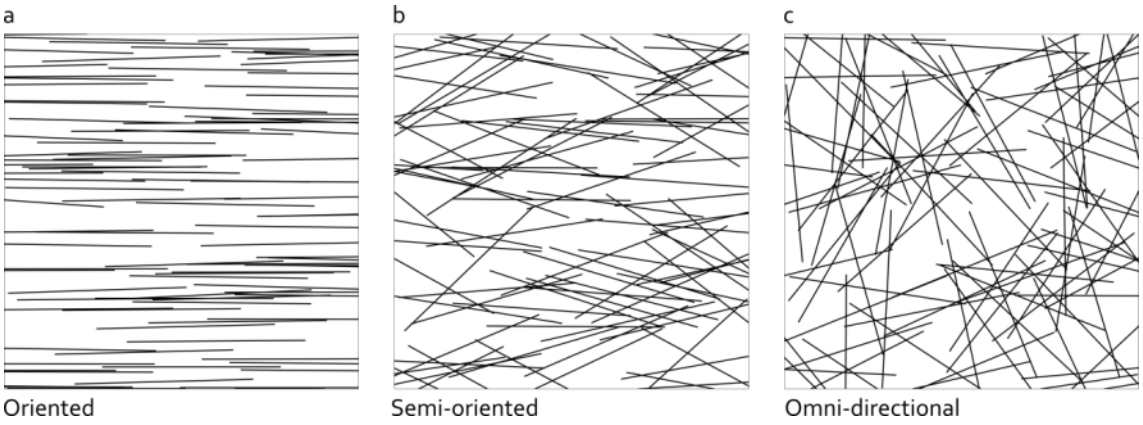


Figure 2.8: Same locations and same lengths but different orientations. In (a) fractures are mainly oriented E-W; (b) partially oriented towards E-W and (c) are randomly oriented. For (a) von-Mises distribution with $mean = 0$ and $\kappa = 1000$, for (b) with $mean = 0$ and $\kappa = 10$ and for (c) with $mean = 0$ and $\kappa = 0$.

Any degree of orientation complexity can be achieved by combining several fracture orientations generated from von-Mises distributions. In the examples in Figure **2.8**, the orientations were defined separately for each set of points. A more complex degree of complexity could be achieved by overlaying the three sets.

Fracture Sets

Fractures can be grouped into sets according to their orientations, lengths and locations. The sets may be defined on the basis of fracture characteristics or may be defined solely for modelling convenience, e.g., by the frequency of occurrence of fractures in directional intervals. Physically meaningful sets may be inferred from the chronological sequence of fracture propagation, by stress field orientations or any other physical or mechanical characteristic. For example, different orientations for fractures may result from different stages of compression or tension stress in specific directions. In addition, for anisotropic rock, fractures can be classified on the basis of their relative locations i.e., spatial settings (see also Dershowitz 1984).

Lengths

Fracture lengths can be modelled by probability distributions including the uniform, Gaussian, power-law, exponential and log-normal distributions, for example (Odling 1992; Gringarten 1997; Ozkaya and Mattner 2003; see also details in Riley 2005; Figure **2.9**). The power-law distribution is a common model for

simulations in DFN. Power-law distributions (in a broader sense including exponential and log-normal distributions) have a very narrow long right tail meaning that there is a very limited chance of long fractures occurring or that shorter fractures are much more likely than longer ones. The likely cause of this is the energy required to propagate fractures in rock masses. Note that basically fracturing occurs when the stress exceeds the strength of the rock releasing the energy. Consequently due to release of energy the system becomes less active. Also note that larger fractures release larger amount of energy, therefore, due to this significant depletion of energy the probability of further large fracturing decreases dramatically.

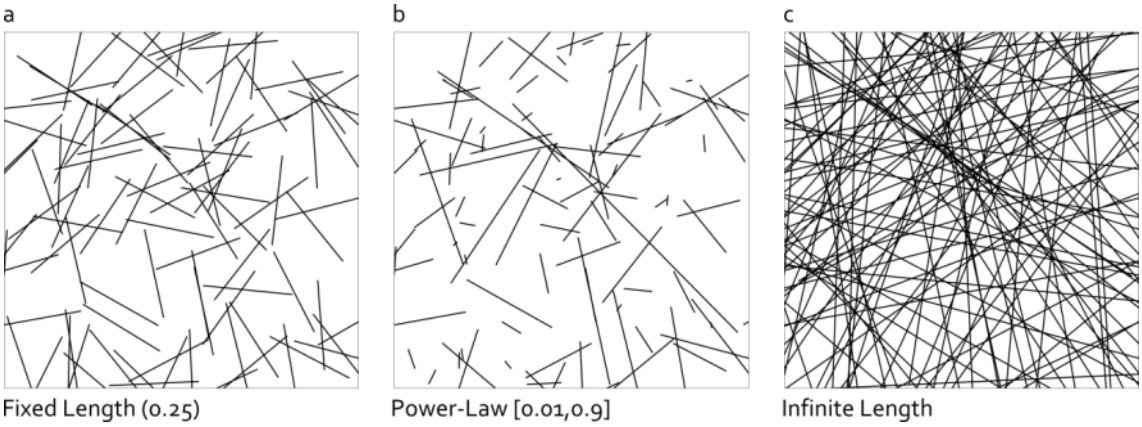


Figure 2.9: Same locations and same orientations but different models of lengths. In (a) fractures are all the same size; in (b) they follow a power-law (exponential) distribution truncated between [0.01, 0.9] and (c) they are infinite in length.

Variations in the patterns of point locations, orientations and lengths generate very different fracture network models, such as those shown in Figure **2.10**. Note that in each example in Figure **2.10** only one fracture set is generated; more complex fracture models can be generated by combining several fracture set models. Figure **2.10** is a demonstration of the ability of DFN modelling to generate significant complexity and variation in fracture network models to satisfy almost any natural situation whilst keeping the modelling procedure straightforward and practical.

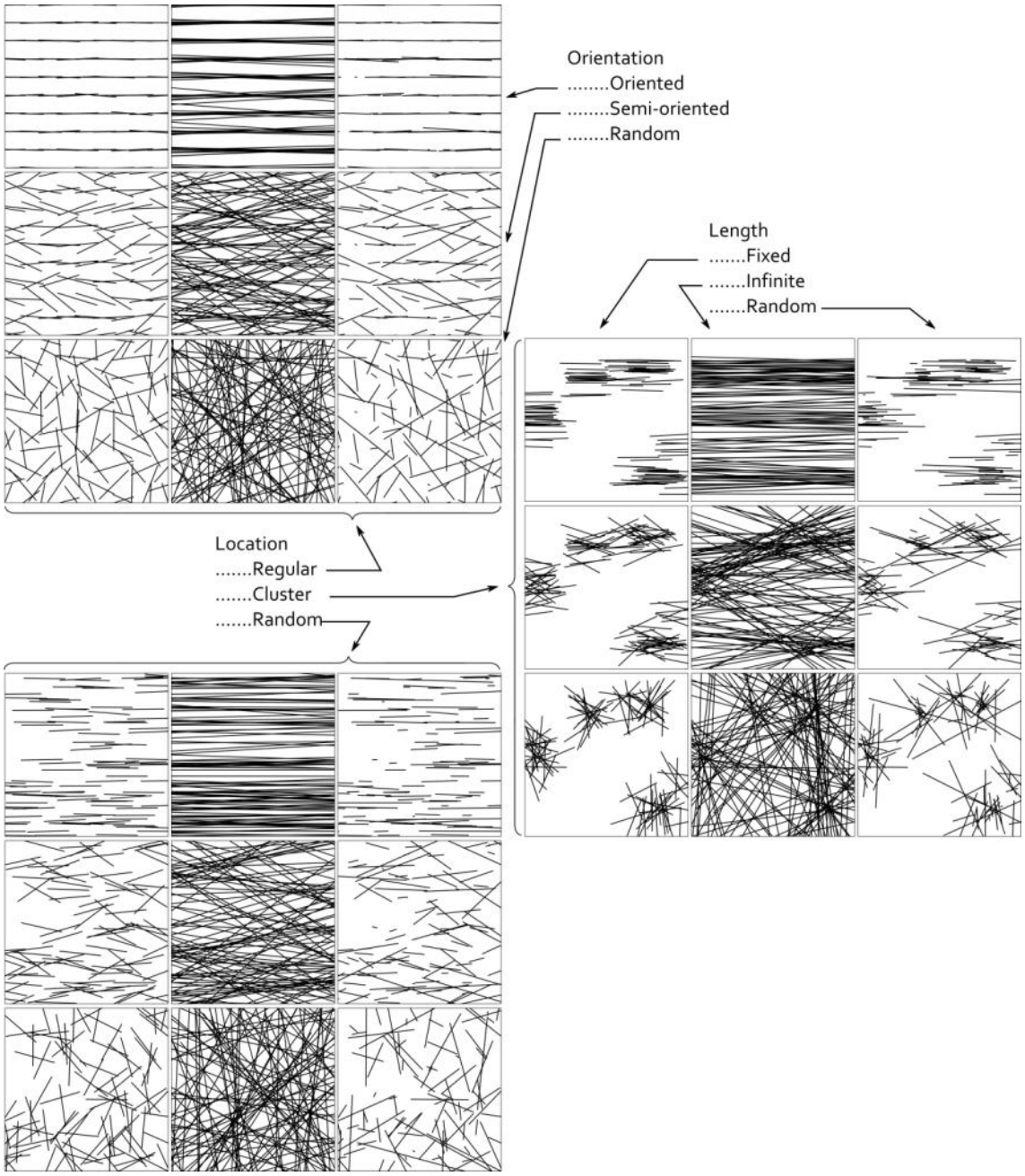


Figure 2.10: Various fracture network models can be generated by combining location, length and orientation models. The examples shown demonstrate the flexibility of the approach to modelling almost any form of fracture network.

2.1.13 Clipping and Cleaning

Fracture network models are often used as inputs to various processes to evaluate their performance under a range of conditions. For example, the potential performance of an enhanced geothermal system could be evaluated by simulating

the flow of fluid through a fracture network model of the system. The ability of fluid to flow through the system is largely a function of the characteristics of the fracture network. Prior to such an application it is common practice to conduct a number of pre-processing operations on the fracture network model including clipping (Goodman and O'Rourke 2004) of fractures at the boundary of the study region (truncating as in Long et al. 1982) and cleaning up fractures resulting from clipping that are shorter than a specified length. If the length distribution function is not truncated at a lower threshold then artificially short fractures may be included in the fracture network model. When a fracture is clipped the vertex lying outside the region is removed and replaced by a new vertex on the boundary, i.e., clipping may generate a fracture that is too short to be consistent with the model. The cleaning up stage removes all artificially short fractures from the model; it is very important for stability and reliability of geometrical operations conducted in subsequent processing applications. As a geometrical operation applied to a FNM of a region $\boldsymbol{\mathcal{R}}$ clipping is defined as:

$$Clip = \mathbf{FNM} \cap \boldsymbol{\mathcal{R}} = \{\mathbf{f} \cap \boldsymbol{\mathcal{R}} : \forall \mathbf{f} \in \mathbf{FNM}\} \tag{2.8}$$

An example of a clipped two-dimensional fracture network is shown in Figure **2.11**. Robust and unbiased statistical analyses require a well-determined study region especially for stochastic modelling in which many realisations will be generated. Generally, as in Figure **2.11** because of the very limited number of fractures being clipped the quality of subsequent evaluations on clipped fracture network including the evaluation of the lengths remains acceptable.

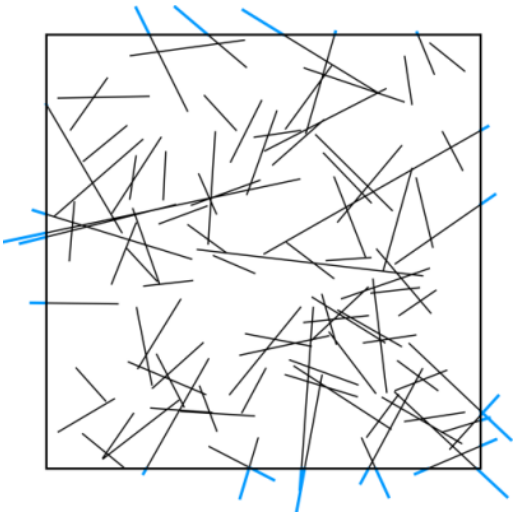


Figure 2.11: Fractures are clipped at the boundaries of the study region. Clipping is followed by a cleaning up stage in which clipped fractures that are too short are removed from the network.

For example, note the updated fracture centres after clipping, shown in Figure **2.12**, which have much the same density distribution (shown as contour map) as the original FNM.

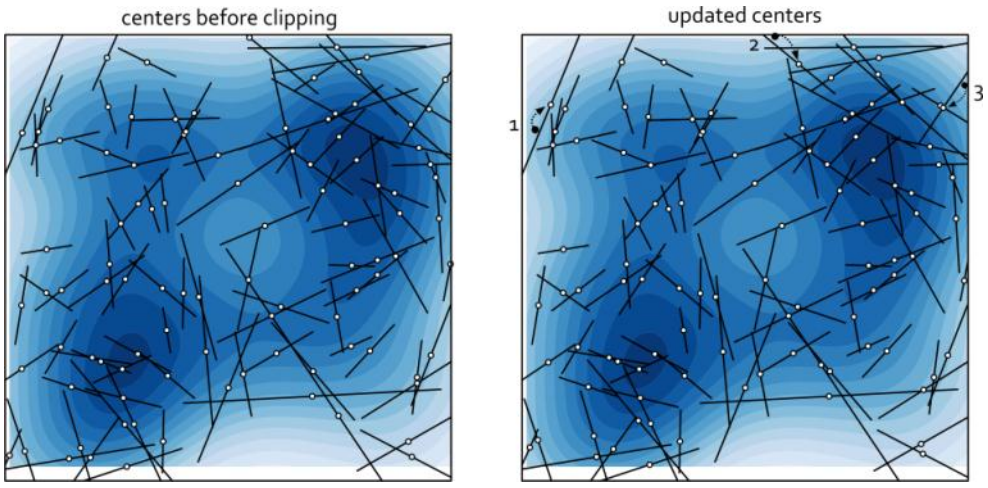


Figure 2.12: After clipping a FNM, the centres of clipped fractures are updated. For this example, updating the fracture centres has no significant effect on the density map of centre points.

2.1.14 Marked Point Processes

Examples of various fracture networks generated by *Marked Point Processes* (MPP) are shown in Figure **2.13**. The demonstrations show how adjustments in the parameters for orientations and lengths (marks) mimic less and more realistic

patterns of fracture networks. The locations in Figure **2.13** were generated by the homogeneous Poisson point process (number of fractures $n = 100$); orientations were produced by the von-Mises distribution (mean orientation μ and dispersion factor κ, varying values as shown in each sub-figure) and lengths were drawn from the truncated exponential distribution ($l \in [0.1, 0.9]$).

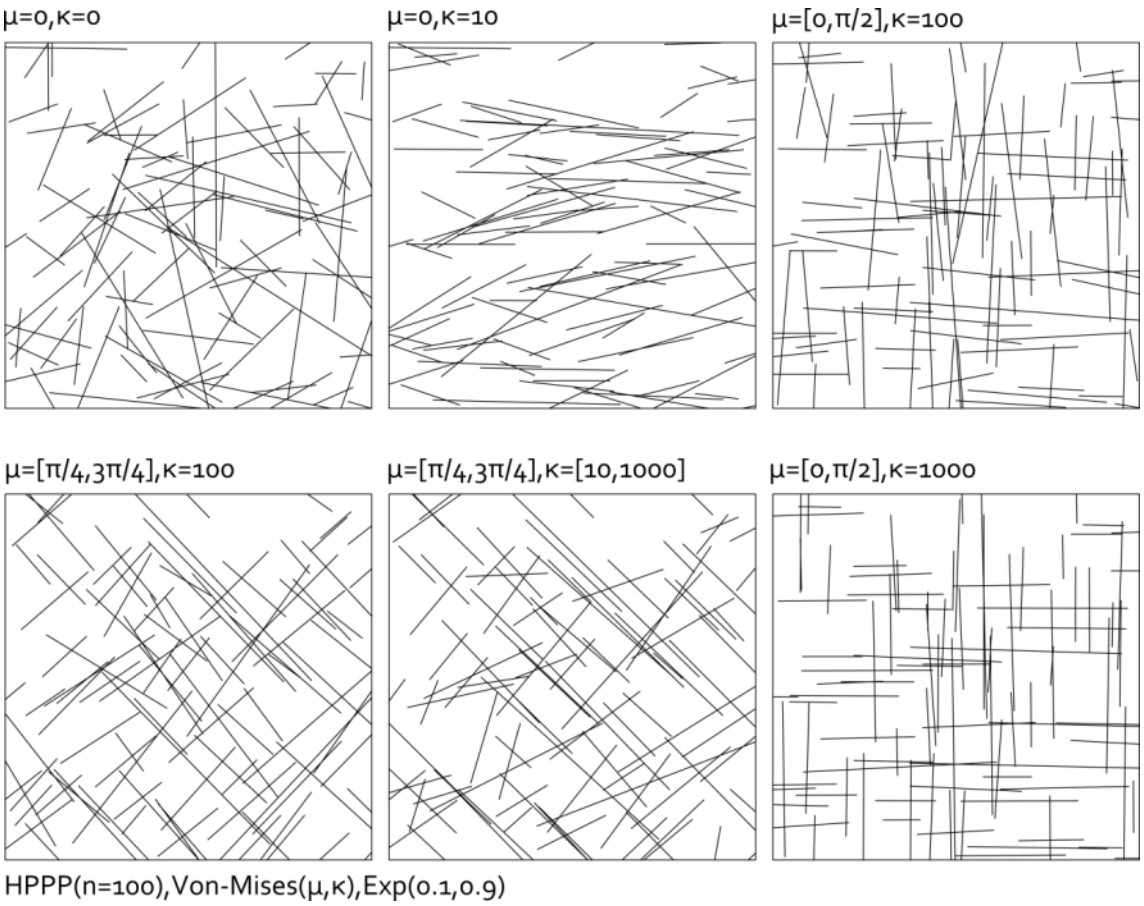


Figure 2.13: Fracture patterns generated by changing parameter values of a von-Mises distribution function for generating orientations of fractures. Pairs of values in captions refer to parameters for two different sets of fractures.

2.1.15 Fracture growth concept

Growing fracture network model (GFNM) is here proposed as a new method for synthesising fracture networks. In this method, the locations of fractures and associated orientations are generated in the same way as in MPP. The lengths of fractures, however, are determined dynamically as follows. Initially, all fractures have lengths equal to *epsilon* ($\epsilon \to 0$), and then all fractures grow simultaneously and at the same rate from both endpoints. The growth of a fracture from each

endpoint stops as soon as another fracture is intersected or at the boundary of the study region (Figure **2.14**). The resulting fracture network produces polygons (*rock mass blocks*, see Goodman and Shi 1985; Stone et al. 1996). The generated polygons (blocks) satisfy the assumptions of *block theory* including 1) fracture (joint) is straight line, 2) there is no incomplete fracture i.e., no free end, 3) blocks are rigid (Goodman and Shi 1985).

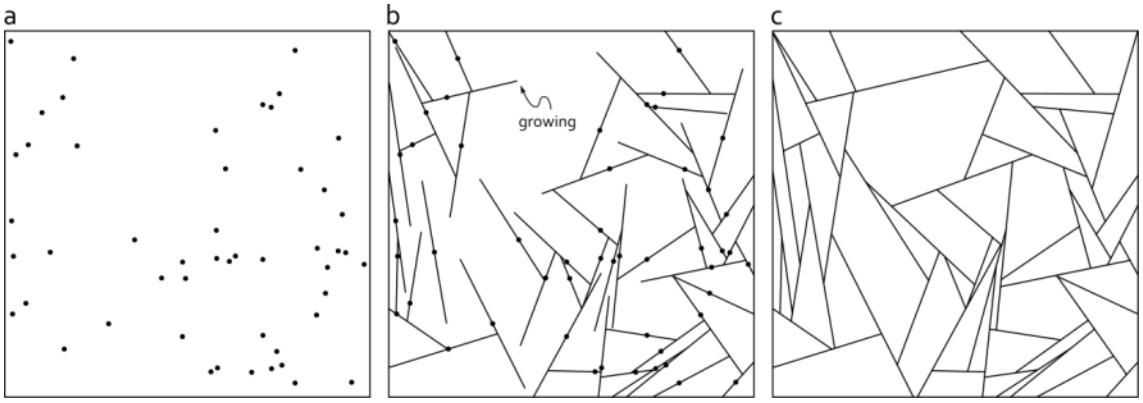


Figure 2.14: Growing Fracture method; (a) 50 locations, (b) some fractures still growing, (c) the final result in which the growing process has stopped and a set of well-defined blocks is created.

Applying the concept of growing fractures with various settings for the fracture network model (e.g., locations and orientations) can generate a wide range of fracture patterns that form rock blocks such as the examples shown in Figure **2.15**. The method provides a useful means of generating blocks that interestingly conforms to field observations.

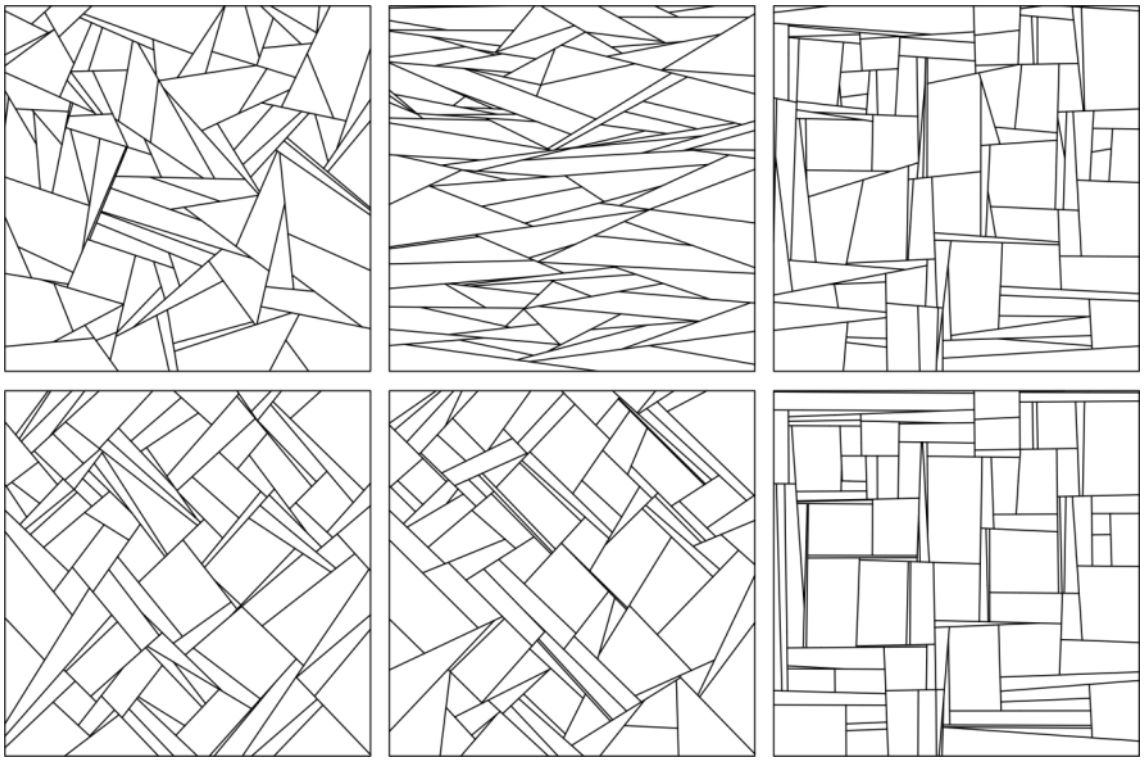


Figure 2.15: Rock blocks generated by means of GFNM for each setting listed in Figure **2.13**.

It is worth noting that, the concept of fracture growth can be readily extended to three-dimensional fractures. It requires, however, more complex geometrical operations in order to apply directional growth to three-dimensional fracture polygons and to evaluate collisions between them. The implementation is mainly a subject of computer sciences, computational geometry.

Growing conditioned to existing fractures

The method of fracture growth interestingly can also honour existing fractures in the study region, i.e., the conditioning of the model to the existing fractures in any amount or complexity requires no further adaptation. This is a valuable property of the method. For example, major fractures (e.g., faults) can be easily taken into account and thus the resulting fracture network (or its dual, the set of fracture blocks) is more realistic (Figure **2.16**).

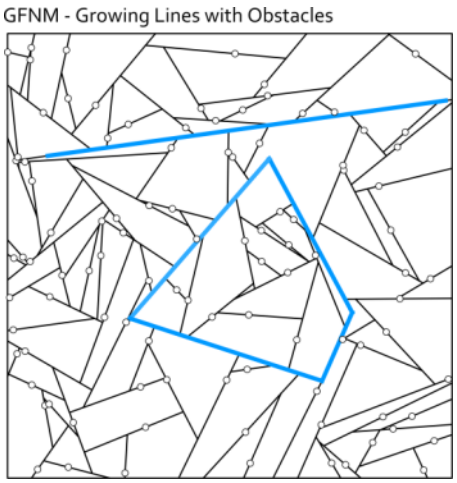


Figure 2.16: GFNM in which growing fractures are conditioned to the existing structures e.g., blue fractures etc.

2.1.16 Voronoi tessellation

Voronoi tessellation, as described in Sect. 2.1.10, is commonly used for generating rock blocks (Figure **2.17**) e.g., in the *UDEC* package (Board 1989; Staub et al. 2002). Although there are many developments in algorithms for computing Voronoi diagrams for various seeds including points, lines and shapes (Okabe et al. 2000), it is very difficult to include conditioning to existing fractures (see Okabe et al. (2000), Gavrilova (2008) and Velic et al. (2009) for proposed pixel-based solutions). Tessellation edge effects are treated using the methods discussed in Sect. 2.1.10, for example.

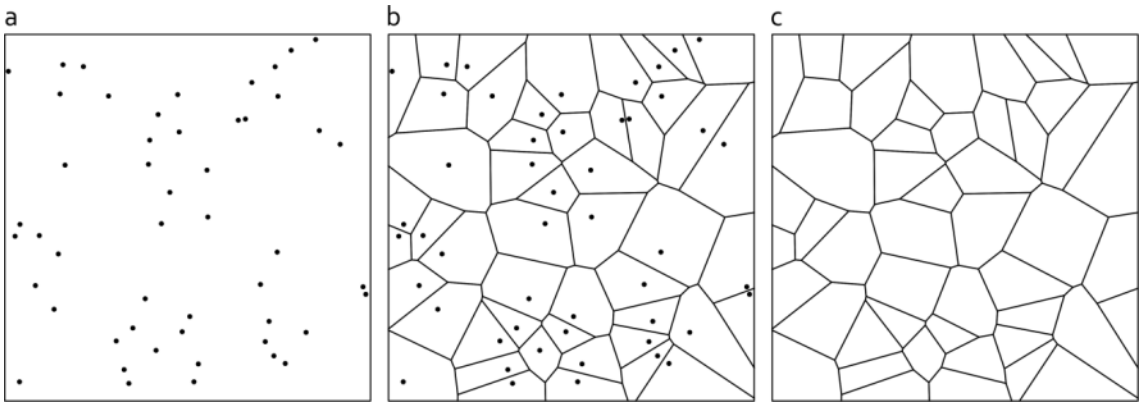


Figure 2.17: Voronoi tessellation fracture network modelling method; (a) 50 locations, (b) Voronoi network and locations (c) the final result in which a set of well-defined blocks is also created.

2.1.17 Block theory

Block theory (Goodman and Shi 1985; Windsor and Thompson 1996) is used in engineering applications, such as mining and civil excavation, largely to assess the safety and stability of structures in rock. It is a geometrically based set of analyses that determines the locations of potentially dangerous blocks of rock intersected by discontinuities, including fractures. The input to block theory analyses is in the form of polygonal shapes (polyhedrons in three dimensions) generated from fracture networks using GFNM, VFNM and other techniques. Stochastic models are useful for generating realistic models for use in block analysis (Dershowitz and Carvalho 1996; Jing and Stephansson 2007a). In addition, the methods presented so far and in the following sections demonstrate the effectiveness of stochastic methods for block theory (see also: Dershowitz and Carvalho 1996; Dershowitz and Busse 1996).

2.2 Additional processing stages for fracture network modelling

Here a set of procedures are proposed that can be applied to a fracture network model to achieve desirable patterns that mimic real fracture networks. For example, a random function can be used to simulate closed (Figure 2.18, filled, for example as a result of precipitation, CFCFF 1996) fractures; a fracture can be removed, shortened, thinned or left untouched on the basis of probabilistic acceptance or rejection criteria. This stage may involve all the fractures in the network or a selected subset of them.

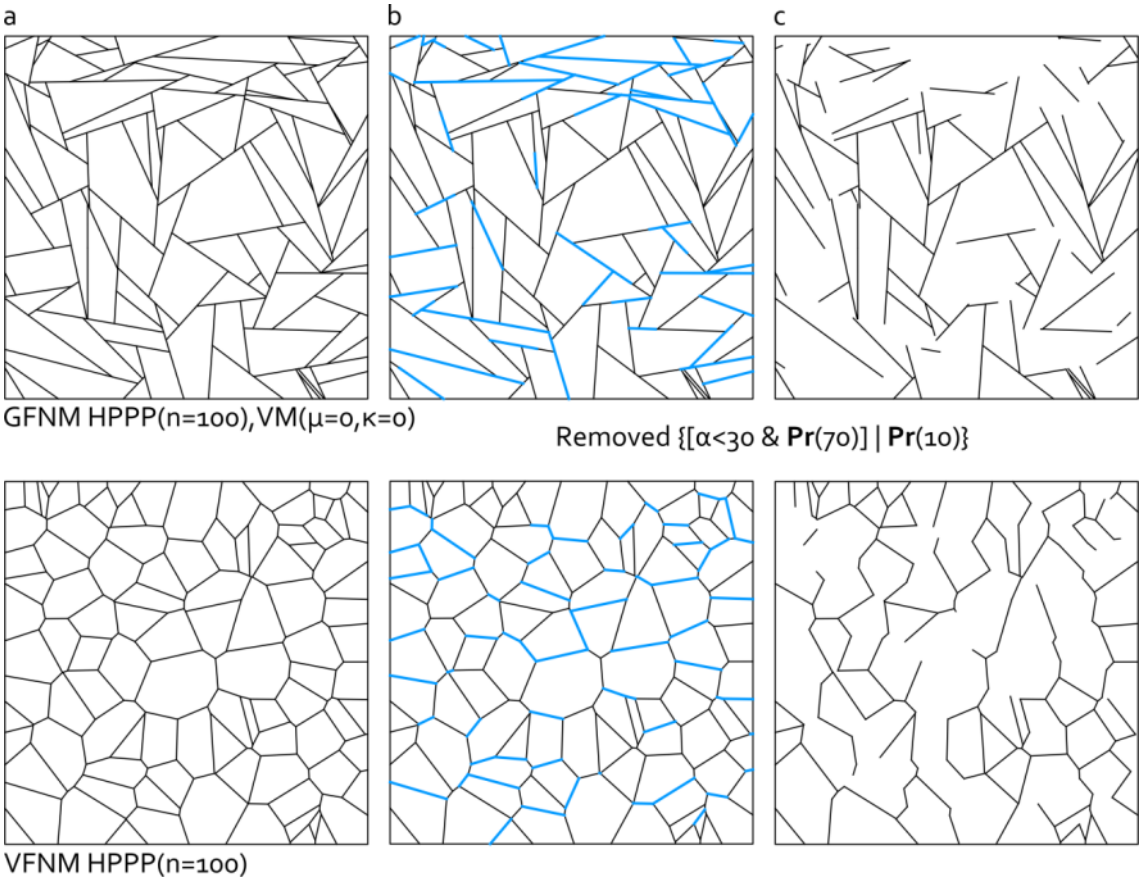


Figure 2.18: Marking a subset of fractures as filled (closed due to precipitation, collapsing stresses etc.), so as to be removed from the fracture network. The criteria for removal is for the acute angle of the fracture to the horizon to be less than 30° and a probability of removal 70%; or 10% probability of removal for any fracture.

2.3 Mathematical Morphology operations applied blocks

As a proposed application the rock blocks generated by the growing concept or by Voronoi tessellation can be processed further using various morphological operations (see mathematical morphology in Kimmel 2003; Illian et al. 2008; see also Sect. 2.1.11). The erosion operation can, for example, be used to simulate the process of geological erosion (e.g., contact surface geochemical dissolution) or geotechnical (mechanical) rock erosion in the fracture network. The thinner rock blocks shown in Figure **2.19** are more affected by the erosion which is consistent with field observations of physical erosion. The morphological erosion procedure applies a negative and positive buffer to each block, sequentially (see details in Sect. 2.1.11). In Figure **2.19** the growing method produces thinner rock blocks

compared to those resulting from Voronoi tessellation. In the erosion operation thinner blocks more likely fully are removed. In reality, thin blocks can be seen in metamorphic rocks such as slates and also in layered sedimentary rocks.

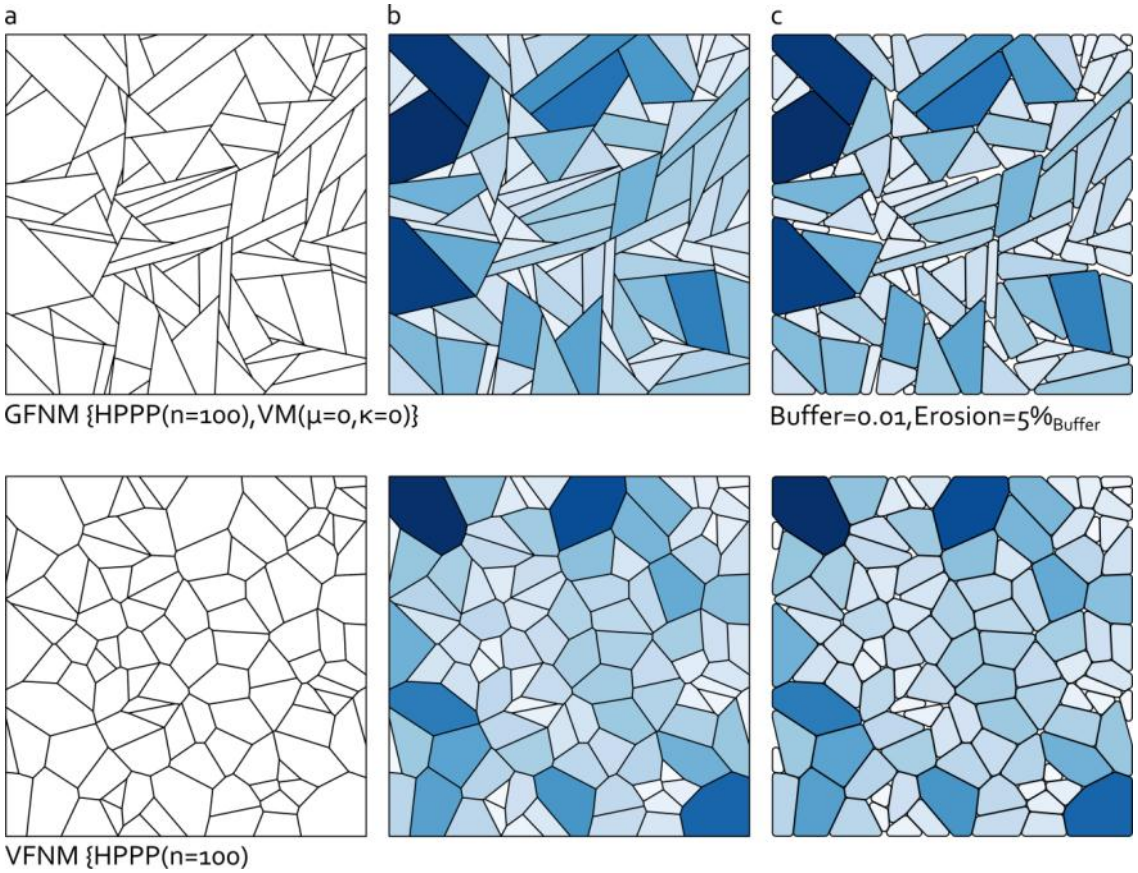


Figure 2.19: Blocks generated by GFNM and VFNM can be subjected to mathematical morphology operations such as erosion. These models simulate rock block erosion due to contact deformation or fluid flow, for example. It can be seen that GFNM is affected more than VFNM by the erosion procedure. The reason is the type of blocks.

Application of morphology concepts to a fracture network model e.g., VFNM as shown in Figure **2.19**, results in a fully connected fracture network in which the evaluation of fluid flow pathways is of interest. Note that the penny-shaped black spots in Figure **2.20** are due to the removal of nearby thinner blocks by the erosion operation.

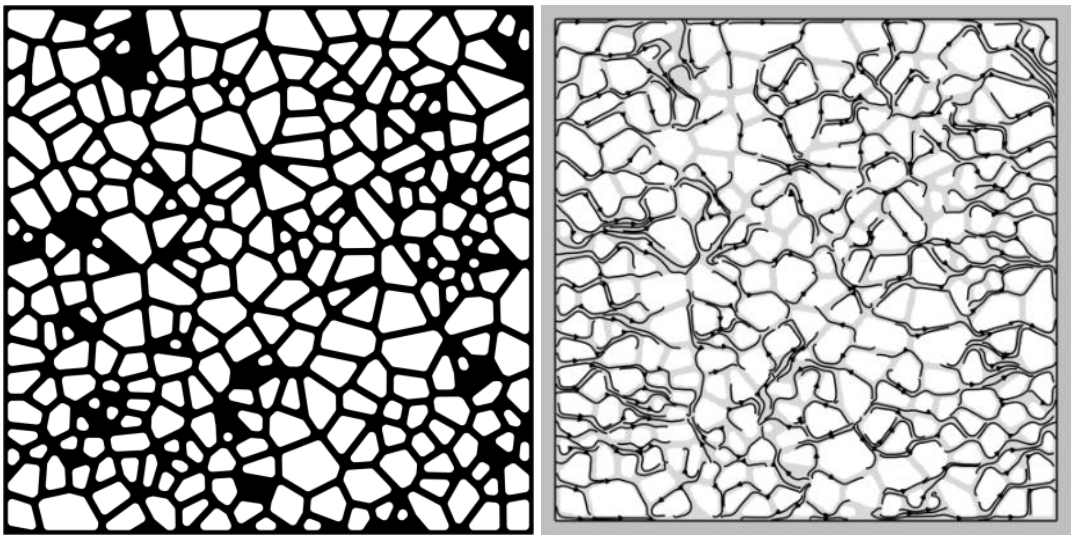


Figure 2.20: Fully connected pathways (left), which can be used as input for evaluation of fluid flow (right) in fractured rock.

2.4 Extension and trimming

The termination of fractures is arguably related to the mechanical evolution of the fracture system (Manzocchi et al. 1998). In MPP models fractures may cross each other i.e., "X type", touch the other i.e., "Y type", or remain isolated, i.e., "I type". The two types X and Y produce blocks, however, type I requires a post-processing stage comprised of an extension and trimming for producing blocks. Figure **2.21** shows the blocks that result from extension and trimming. An endpoint of a fracture trace line is extended or trimmed depending on the cost (energy consumption) involved. It is assumed that the same amount of energy is required for one unit of extension or trimming. Note that in this method, denoted *TFNM*, some resulting blocks may be concave in contrast to GFNM and VFNM in which blocks are always convex. As in reality not all blocks are convex the resulting blocks of TFNM are therefore more realistic.

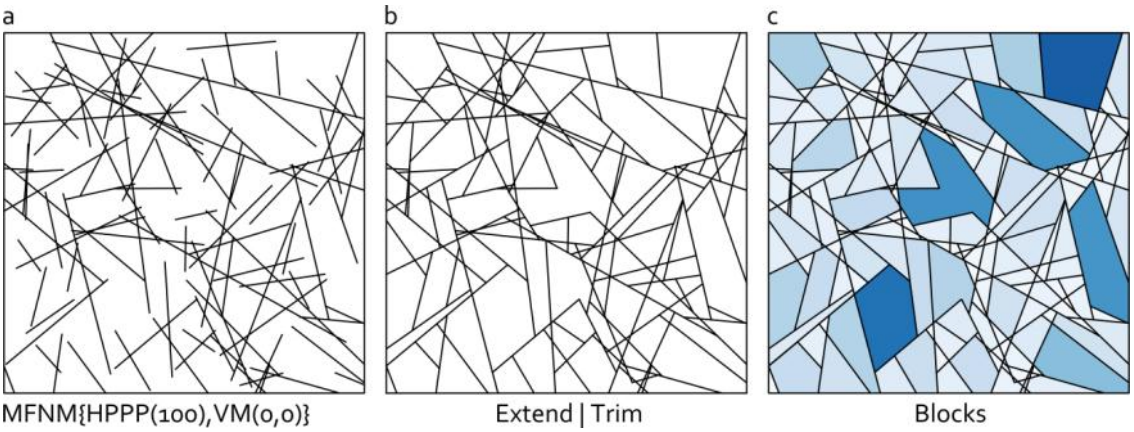


Figure 2.21: Blocks generated by extension and trimming. Fractures are extended or trimmed depending on the cost of extension or trim. This approach may result in convex or concave blocks.

2.5 Three-dimensional modelling of fracture networks

In three dimensions a fracture is better represented by polygon in its generic meaning providing convenient way to model any complex and unsymmetrical shapes. Other methods including circle, ellipse, rectangle and plane representations are too simplistic. In modelling methods, such as DFN, polygons are strictly convex so as to avoid unnecessary complexity in analytical and numerical solutions. This condition does not limit the modelling capability of the method as any concave polygon can be readily sub-divided into convex polygons. Restricting the polygonal representation to convex shapes also results in a robust framework for topological (connectivity) relationships in fracture networks. Furthermore, despite the apparent complexity of a general convex polygon compared to a circle, ellipse or rectangle, the geometrical operations are straightforward if not more efficient (see Goodman and O'Rourke 2004). For example, joining a set of topologically ordered points (forward joining) defines a convex polygon using the method demonstrated in Figure **2.22**. In this proposal either the enclosing domain is a circle or an ellipse, a set of random values in [0, 2π) (angles) are generated and sorted. The vertices of the polygon are then determined by lines of orientation α and lengths equal to the chords these lines make with the circle or ellipse.

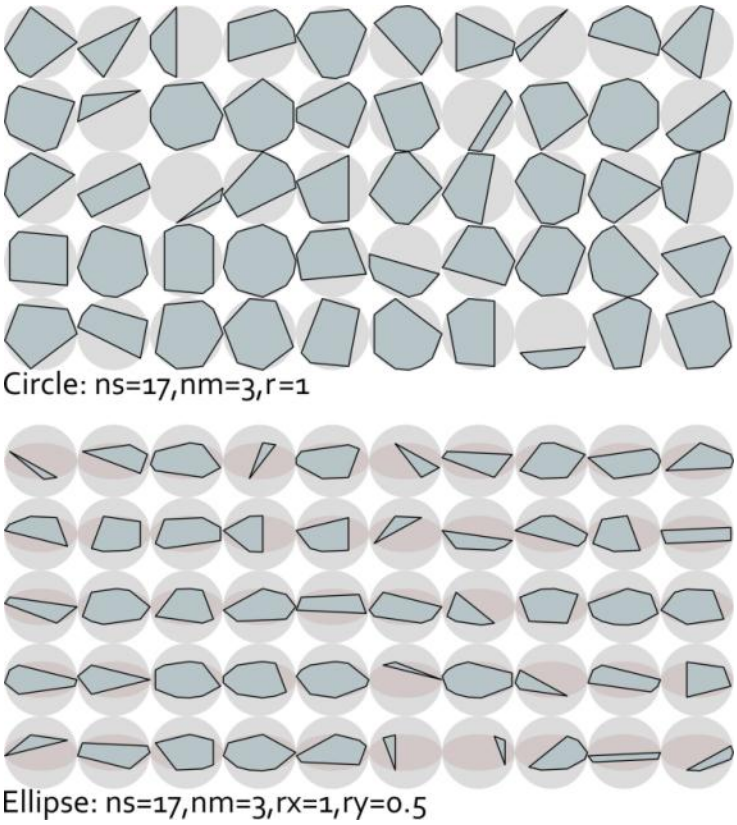


Figure 2.22: A simple but robust method to generate polygonal fracture shapes for three-dimensional fracture network simulations. The method is based on generating convex polygons enclosed by a circle or an ellipse. As shown the latter provides a means of accounting for anisotropic shapes. The variety of the shapes in both examples demonstrates the ability to accommodate almost any polygon. Also note the size variation in the second method (ellipse).

The next stage is to generate locations and orientation angles to be applied to simulated fracture polygons. Fracture locations can be generated by a Poisson point process (see Sect. 2.1.3 and 2.1.12), for example. Orientations are generated from a Fisher distribution (Sect. 2.1.12). That is, setting the κ parameter in the distribution affects the dispersion of fracture orientations around a chosen mean orientation vector in a three-dimensional fashion. It is worth noting that a bivariate Gaussian distribution of points if projected on the surface of a sphere generates exactly what Fisher distribution produces. By translating the **CoG** of fracture polygons to the simulated location points and applying three-dimensional rotations according to simulated angles a three-dimensional fracture network is built. An example three-dimensional fracture network (with 150 polygonal

omnidirectional fractures) as a result of application of described method is shown in Figure **2.23**a.

It is also worth noting that, many of developed and proposed ideas in this thesis can be applied either straightforwardly or with some efforts on three-dimensional fracture networks. This is not however the focus of this thesis by any means, although, time to time some recommendations are given. For reader's convenience I have appended a full software package called Alghalandis Fracture Network Modelling (AFNM) which was developed in Matlab environment. It can be used to conduct easily many of the ideas discussed in this thesis.

2.5.1 Fracture network as pipe model

Most three-dimensional fracture networks are too complex for input to standard finite element methods of modelling fluid flow. The example fracture network shown in Figure **2.23**a gives an indication of the level of complexity of a fracture network that would have to be meshed (discretised) in preparation for modelling by finite element methods. Note that the example shown is a relatively simple network compared to those that could be generated by stochastic modelling. Several proposals are given in the literature to overcome this problem with the primary objective being to simplify the modelling stage. One such proposal is the pipe model (La-Pointe and Wallmann 1996; Dershowitz and Fidelibus 1999; Xu et al. 2013a). In this model, for two connected fractures two pipes are used to connect the centres of the fractures to the centre of intersection. In this way, the apertures of the fractures can be individually represented by the radius of the associated pipe. The pipe model for the example in Figure **2.23**a is shown in Figure **2.23**c.

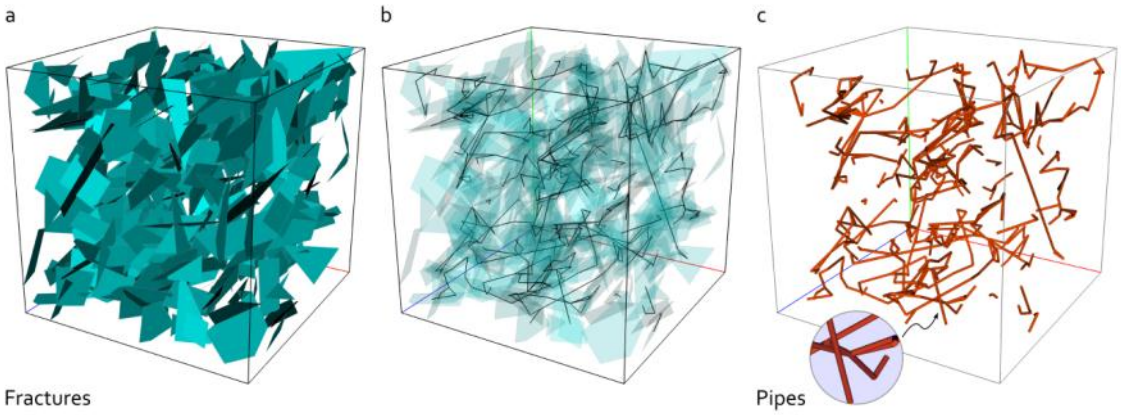


Figure 2.23: Three-dimensional fracture network and associated pipe model. Pipes are generated by connecting the centres of each fracture to the centre of intersection line (point in vertex touching case). The radius of pipes can be defined locally according to the apertures of the associated fractures.

2.6 Conditional fracture network modelling

Two types of conditioning are applicable in fracture network modelling: 1) Intrinsic conditioning in which the model honours any given information by adapting the distribution functions which are used for attributes (marks). This method can also be seen as pre-conditioning; 2) The resulting fracture network model can also be adapted to satisfy given conditions such as regenerating specified existing fractures in the study region. Whilst the latter may appear more interesting in terms of real applications, it is important not to disturb the stochastic spatial features of the simulated fracture model when conditioning it to data or prior information.

2.6.1 Conditioning fracture locations using Simulated Annealing

Simulated Annealing (**SA**, Deutsch and Cockerham 1994; Tran 2007) can be used to manipulate a FNM simulation to honour any existing data e.g., sampled fractures. It is an optimisation method based on perturbation of parameters to minimise a specified cost function. The proposed example in Figure **2.24** shows the results of applying SA during the generation of a fracture network for an entire study region to condition it to 20 existing fracture locations (hard data) within an area equal to 1/5 of entire region. The resulting conditioned locations are shown in Figure **2.24**d and a comparison is made in Figure **2.24**e of locations before and after

the application of SA. In this example, SA reaches a stable state after around 2000 iterations. A plot of the evolving objective function, (OFn), is shown in Figure **2.24**c. Here OFn is defined as a function of the number ($\langle N \rangle$) of points in the conditioning area and their distances (D) from conditioning data.

$$OFn = |\langle N \rangle - n| + \sum \min_{i=1..n} D_i \tag{2.9}$$

where n is the number of hard (conditioning) data.

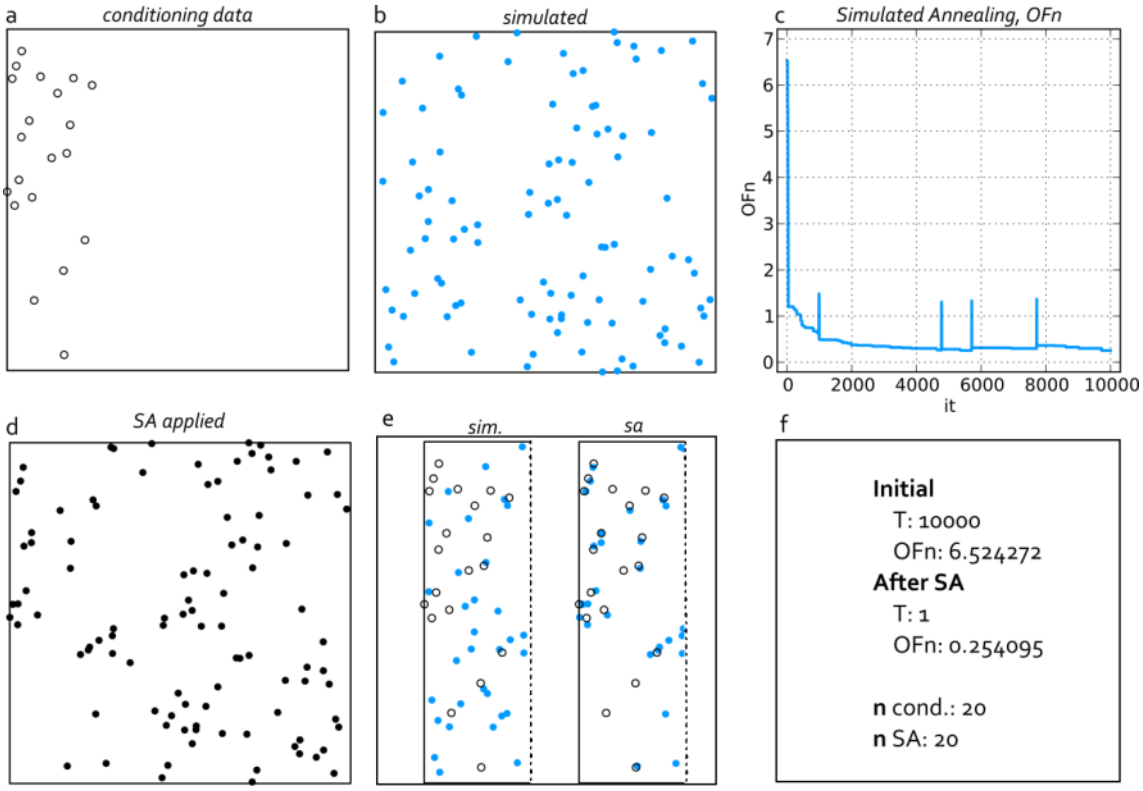


Figure 2.24: Conditioning fracture locations to given sample points. Simulated Annealing is used to honour given locations.

2.6.2 Conditioning to existing fractures

When fracture observations and measurement are available (e.g., measured on drill cores, borehole walls, tunnel walls or outcrops), they can be used to condition the simulation. If the independency assumption in DFN is accepted (i.e., all fractures are generated independently of all others), then it is a simple task to generate conditional fracture networks. The solution, as proposed by Andersson et al. (1984)

and Andersson and Dverstorp (1987) and developed further by Chilès and de-Marsily (1993), is: 1) generate an unconditional simulation of the fracture network; and 2) remove any fracture that intersects the conditioning area e.g., borehole, tunnel walls etc. The resulting fracture network is thus conditioned to include all observed fractures whilst preserving the unconditionally simulated fractures in the rest of the study region. This simple method assumes that the conditioning area has been exhaustively sampled for fractures, which for two and three dimensions may not be the case. It also creates significant edge effects around the conditioning area and the conditioning area biases true fracture lengths. Figure **2.25** shows a fracture network in two-dimensions covering an entire study region conditioned to fractures observed in two boreholes.

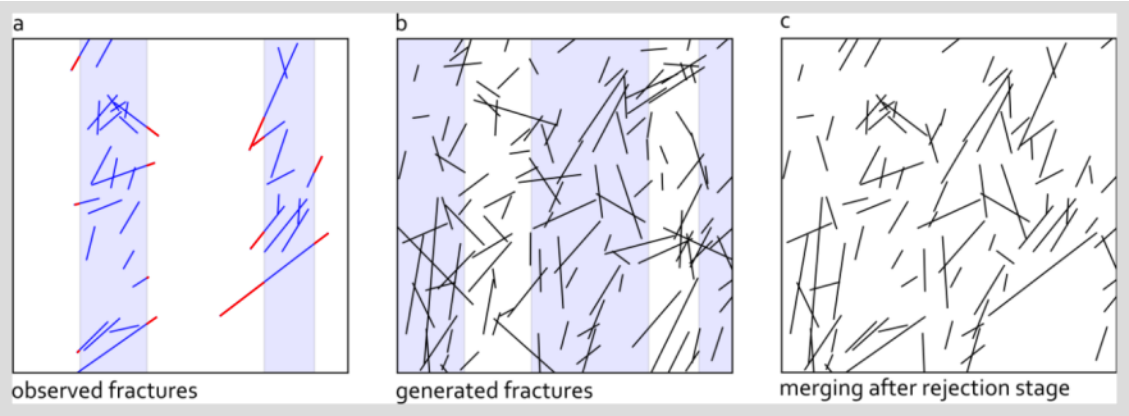


Figure 2.25: Conditioning fracture network to existing fractures observed in two boreholes.

In the demonstration given in Figure **2.25**a the fractures (blue) observed in the boreholes are clipped by the borehole perimeter. One may use a suitable distribution function such as Gaussian in order to simulate full length of fractures. As the rejection criteria in the above conditioning method only applies on borehole boundaries such an expansion has no effect on simulation but has two benefits: 1) the unobserved parts of longer fractures are being modelled rather than being ignored 2) boundary artefacts in the proximity of borehole perimeter is decreased. The described conditioning can be applied at the beginning or at the end of the modelling procedure. The conditioning technique described is applicable to three-dimensional fracture networks, in which fracture polygons can be generated in such a way as to avoid intersecting any conditioning volumes, such as boreholes. A further development of this conditioning method would be to control the fracture

density in the generated network to follow specified distributions or patterns. To do so, density evaluation may apply on conditioning area resulting in information for the density of entire study region. If the density map of the conditioning area shows specific pattern / structure it may be desired somehow to generalise the structure onto the entire simulated region.

2.6.3 Conditional simulation to a point cloud

Hough transform

The detection of objects from point data is an active research area in computer vision and has many applications in other disciplines. In fracture network modelling for enhanced geothermal systems, for example, micro-seismic events generated by fracture stimulation of the reservoir provide data in the form of a point cloud (i.e., set of points). A micro-seismic event is associated with a movement in the rock caused by the hydraulic pressure of the fluid injected into the reservoir and distributed via the extension of existing fractures and newly propagated fractures. The locations of the events are determined by processing the recorded seismic logs. Note that no topology information is detected or recorded during the procedure. That is, the resulting is in the form of points distributed in the space often without any apparent structure or alignment.

The Hough transform (HT, Hough 1962; Duda and Hart 1972) is a simple transform which projects any point in Euclidean space onto a pair comprising an angle and a distance (parameter space). The line equation in Euclidean space is:

$$y = ax + b \tag{2.10}$$

It can be easily rearranged into parameter space as:

$$b = -xa + y \tag{2.11}$$

Note that in Eq. 2.10 the parameters a and b are constant for any particular line. Therefore in its transformed form (Eq. 2.11) any point (x, y) on the line will generate the same a and b parameters. This is called a "voting" procedure in which

the pairs of parameters (a, b) corresponding to the highest votes relate to the lineaments in the point data in Euclidean space. This procedure is called *line detection*. An example application of HT to fracture network modelling (a proposal) is shown in Figure **2.26** in which Figure **2.26**b is a point cloud in two dimensions prepared by discretisation of the fracture network shown in Figure **2.26**a. Additional random points completes the generation of point cloud. The aim is to see whether HT can detect embedded lines in the point cloud. The parameter space is shown in Figure **2.26**c which is an accumulation of sinusoids $(\rho = x \sin\theta + y \cos\theta)$ where θ is an angle in degrees and ρ is distance in pixels. Red pixels denote the highest votes corresponding to their dual lines in Euclidean space as shown in Figure **2.26**d.

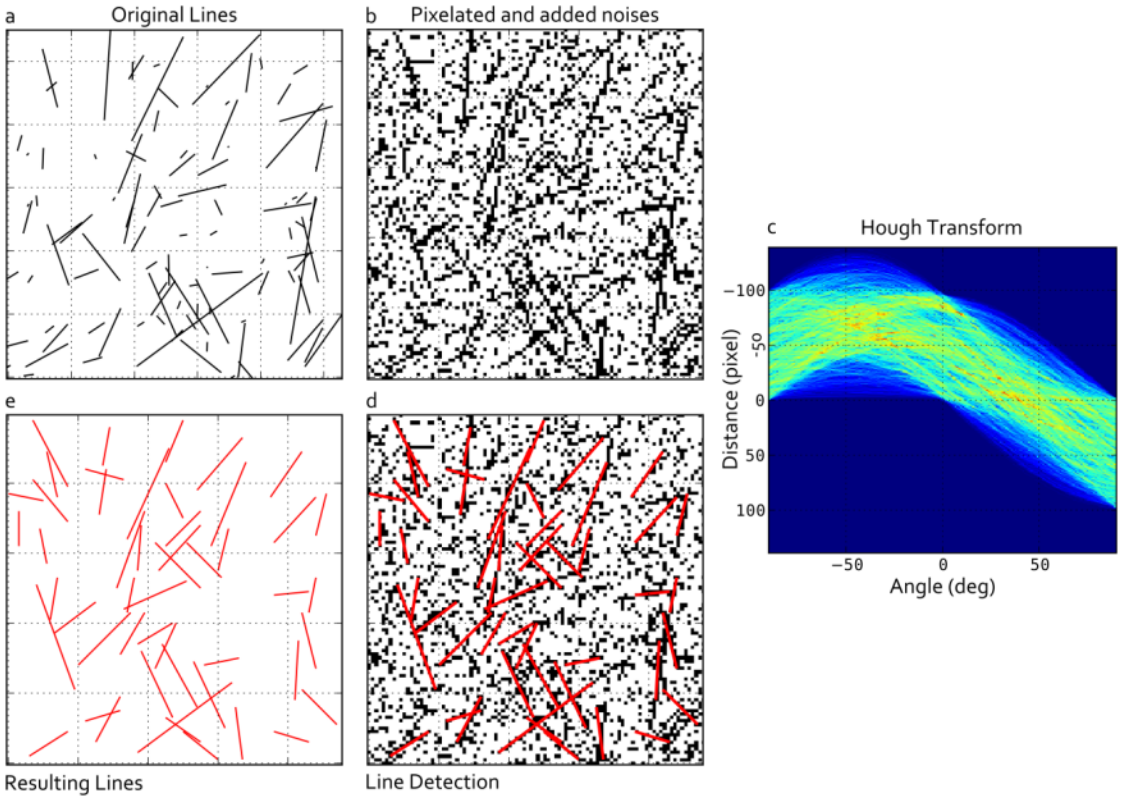


Figure 2.26: Line detection using the Hough Transform. In (b) the generated lines in (a) are discretised into pixels added significant noise. The Hough transform for (b) is shown in (c) which results in linear objects shown as red in (d). The resulting lines (e) compared to the original lines (a) show a very good match despite the effects of noisy data added in (b).

Comparing the results in Figure **2.26**e with the original embedded lines in Figure **2.26**a it can be concluded that HT works satisfactorily when applied to

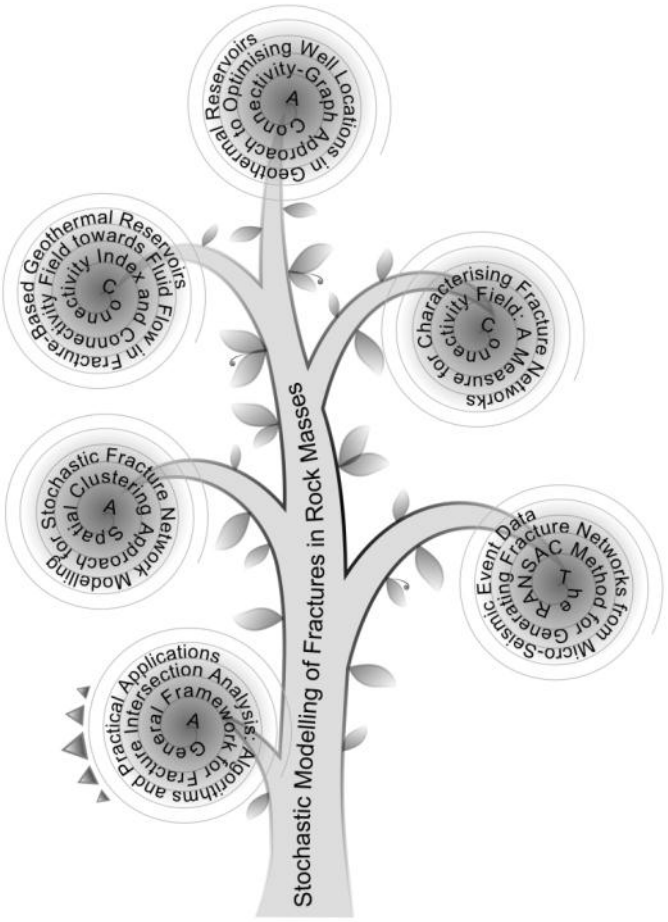
highly randomly populated point cloud. A similar concept can be developed for three dimensional fracture networks in order to detect planar shapes in point clouds.

RANSAC

Random Sample Consensus (RANSAC) is an algorithm for robust fitting (matching) of models to data that include significant numbers of outliers. In the presence of outliers RANSAC is significantly more robust than the traditional least squares methods. It can be used for extracting possible fractures lines or planes from micro-seismic point clouds, for example. The RANSAC method is very efficient in dealing with large datasets. Chapter 4 comprises a comprehensive, published paper on RANSAC and its application to fracture network modelling.

3

A General Framework for Fracture Intersection Analysis: Algorithms and Practical Applications



In Chap. 2 fracture network modelling was studied. Details of new proposals were presented. In the present chapter which is a published conference paper a couple of additional fundamental discussions on fracture network modelling are presented. Some new measures are introduced that help to characterise fracture networks. These include intersection density, distribution of length of fracture intersection lines in a fracture network and the effect of fractures on the percolation state of two-dimensional fracture networks. As shown in the illustration above this paper is the first paper that started building the publication structure of this thesis. The tree shows major published papers from this research in a chronological order from bottom to top.

Statement of Authorship

Title of Paper	A general framework for fracture intersection analysis: algorithms and practical applications.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	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, p15-20.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	proposing and development of the concepts, conducting simulations and data analysis, developiong algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to the concepts and models on which the paper is based as reflected in the references listed in the paper. Co-supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

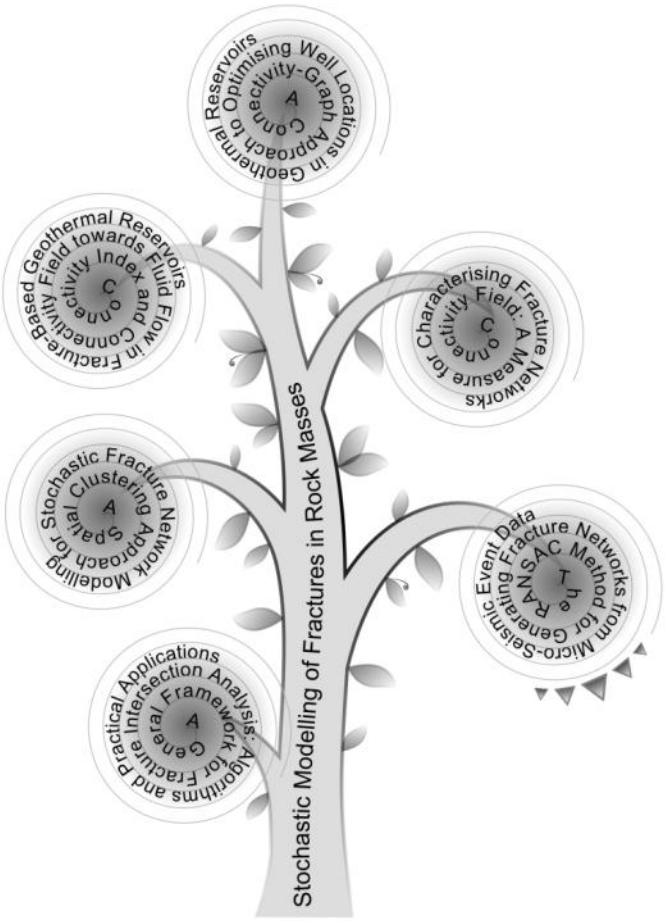
Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to the concepts and models on which the paper is based as reflected in the references listed in the paper. Co-supervising development of work, manuscript evaluation and extensive editing.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

Fadakar Alghalandis, Y., Xu, C. & Dowd, P.A. (2011) A general framework for fracture intersection analysis: algorithms and practical applications.
Presented at: Australian Geothermal Energy Conference, Melbourne, Australia, pp. 15-20

4

The RANSAC Method for Generating Fracture Networks from Micro-Seismic Event Data



From the discussions in Chap. 2 and 3 the present chapter, which is a published journal paper, focuses on the conditional modelling of fracture networks. The conditioning is applied in such a way that the generated fracture network honours the existing point cloud data which are collected during the stimulation process in EGS development. This paper contributes to the fracture network modelling stage of the research and also introduces for the first time the use of the RANSAC method in fracture network modelling. Other areas of related disciplines may benefit from this contribution.

Statement of Authorship

Title of Paper	The RANSAC method for generating fracture networks from micro-seismic event data.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Fadakar-A Y, Dowd P.A, Xu C (2013) The RANSAC method for generating fracture networks from micro-seismic event data. J Mathematical Geosciences, 45:207-224.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	Proposing and development of the concepts, conducting simulations and data analysis, developing algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to the development of concepts, supervising development of work, manuscript evaluation and significant editing and rewriting of the manuscript.		
Signature		Date	28/2/2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to the development of concepts, co-supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

The RANSAC method for generating fracture networks from micro-seismic event data

Younes Fadakar Alghalandis[1], Peter A. Dowd[2], Chaoshui Xu[3]

Abstract Fracture network modelling is an essential part of the design, development and performance assessment of Enhanced Geothermal Systems. These systems are created from geothermal resources, usually located several kilometres below the surface of the Earth, by establishing a network of connected fractures through which fluid can flow. The depth of the reservoir makes it impossible to make direct measurements of fractures and data are collected from indirect measurements such as geophysical surveys. An important source of indirect data is the seismic event point cloud generated by the fracture stimulation process. Locations of these points are estimated from recorded micro-seismic signals generated by fracture initiation, propagation and slip. This point cloud can be expressed as a set of three-dimensional coordinates with attributes, for example, $Se_{ijk} = \{(x, y, z;\ a) | x, y, z \in R, a \in I\}$. We describe two methods for reconstructing realistic fracture trace lines and planes given the point cloud of seismic events data: Enhanced Brute-Force Search and RANSAC. The methods have been tested on a synthetic data set and on the Habanero dataset of Geodynamics' geothermal project in the Cooper Basin of South Australia. Our results show that the RANSAC method is an efficient and suitable method for the conditional simulation of fracture networks.

Keywords Fracture Network Modelling, Line/Plane Detection, Point Cloud, RANSAC, Conditional Modelling.

1 Corresponding Author: Younes Fadakar Alghalandis. School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, 5005, SA, Australia, Phone: +61 8 8313 1570 Fax: +61 8 8313 4359 emails: yfa.study@ymail.com or younes.fadakar@adelaide.edu.au

2 Faculty of Engineering, Computer and Mathematical Sciences, The University of Adelaide, Adelaide, SA, Australia

3 School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, SA, Australia

4.1 Introduction

The technical and commercial viability of producing geothermal energy from hot dry rock (HDR) depends on the creation of artificial reservoirs, or Enhanced Geothermal Systems (EGS), in the rock mass by stimulating and creating fractures (generally by hydro-fracturing) to enable geothermal flow. The artificial reservoir is the critical component of an EGS: the fracture network that connects the injection and production wells and acts as the heat exchange chamber for the system. The productivity of the system depends crucially on the connectivity/permeability of the reservoir fracture network and a realistic and reliable fracture model is the key to assessing reservoir performance and designing a suitable heat exchange chamber for the EGS.

The characterization of rock fracture networks is a very difficult problem not least because accurate field measurement of a single fracture is difficult and measurement of all fractures is impossible. Thus, in practice, the whole fracture system is not observable on any meaningful scale and the only realistic approach is via a stochastic model informed by sparse data and/or by analogues. In HDR applications, a realistic solution is even more difficult as the only reference data related to the fracture system are from limited drill core samples, geophysical borehole logs or sparse seismic events kilometres beneath the surface detected during the hydraulic stimulation process. During fracture stimulation, fracture initiation, propagation and slipping generate micro-seismic events that can be monitored and their spatial locations determined. The resulting seismic point cloud can be used to determine the geographical extent of the HDR reservoir, the hydraulic performance, the amount of fracturing and the fracture network within the reservoir (Shapiro et al. 1998; Audigane et al. 2002; Bruel 2007; Xu et al. 2011; Li and Anderson-Sprecher 2011). Post-processing of the seismic data yields a data set comprising three-dimensional coordinates of the events and attributes inferred from the seismic signals.

Assuming that fractures can be modelled by lines in two dimensions or planes in three dimensions (Fadakar-A et al. 2011; Toth and Vass 2011), the problem addressed in this paper is the assignment of lines or planes to the seismic event

locations to provide an *optimal* realization of the fractures represented by the micro-seismic events. Any three seismic locations define a plane that may be a candidate for a fracture surface. As the number of points (locations) used for fitting a surface increases, the shape of the inferred fracture surface changes as a varying convex-hull. This can be envisaged as a three-dimensional envelope that includes all points within some specified distance from the central plane fitted to the initial three points. The computational difficulty of fitting fracture surfaces increases as the number of points in the point cloud increases and the dispersion pattern of the points becomes more complex. This is most apparent when attempting to fit the best plane through the entire point cloud.

In the following sections we discuss the difficulties of, and some solutions for, generating fractures in the form of lines from a two-dimensional point cloud and polygons from a three-dimensional point cloud. An attempt to investigate relationships between seismic events and the occurrence of fractures in an EGS has been reported in Xu et al. (2011). The technique used was Markov chain Monte Carlo (MCMC) simulation, which in general incurs a very high computational cost. The methods reported in this paper are very computationally efficient which makes them more advantageous in dealing with large data sets.

4.2 The Problem of Fitting Lines/Planes to Point Cloud: Conditional Simulation

There is a significant body of work reported in the computer vision literature and in related areas such as image processing and intelligent robotics, dealing with the detection of objects from input data sets, for example, photographic images or automatic sensing systems (Duda and Hart 1972; Roth and Levine 1993; Hofer et al. 2005; Szeliski 2010; Yang and Wolfgang 2010). The input data essentially comprise a collection of points (or small regular shapes, such as pixels, that can be represented as points). A collection of such point data is often called a point cloud. Often there is no direct topological information about the points and it must be inferred by some form of modelling or matching.

We introduce the Random Sample Consensus (*RANSAC*) method for extracting possible fractures lines or planes from micro-seismic point clouds. In this paper we focus on adapting RANSAC methods for our particular application. Our study uses two sets of data: stochastically simulated point clouds (two- and three-dimensional) and a real data set of seismic events in an EGS. Note that in both cases the reconstruction of fracture lines/planes is analogous to the conditional simulation of an FNM (Cherpeau et al. 2012) conditioned to the seismic events point cloud observed in the field. The proposals presented here have potential application to other types of conditioning data.

4.2.1 Enhanced Brute-Force Search

Brute-Force Search (BFS) is an exhaustive search method that finds the best solution after examining all possible solutions (Harris and Ross 2005). The method can be applied as a single exhaustive search or as an iterative search. Although BFS is an intuitive and very simple approach we detail here the difficulties in dealing with large datasets. This section also proposes a set of improvements and provides a logical move to, and a background for, the RANSAC method in the next section.

Fitting a fracture network to a set of points requires a distance (d) tolerance (τ) to assign points to candidate fractures. A criterion for the goodness-of-fit of a fracture to a set of points is to maximize the number of nearby ($d \leq \tau$) points associated with the candidate fracture. In other words, the goodness-of-fit increases as the number of points associated with a fracture increases. This criterion is used to rank all candidate solutions and select the best. The steps in BFS are as follows.

1. Select two points sequentially from the point cloud and define a candidate fracture line $l_c = \{(p_1, p_2)\}$
2. Compute the rank r_c of the candidate line, which is a count function of nearby points with distance ($\mathbf{d}_{ic}$) less than a distance tolerance tol thus $r_c = \#(\mathbf{d}_{ic} < \tau) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{1}(\mathbf{d_{ic}} < tol)$
3. Store the rank and go to step 1

4. Sort the ranks in descending order and choose a specified number as the candidate fracture lines

Figure **4.1** demonstrates the application of BFS to a small artificial two-dimensional point cloud comprised of 367 points (300 uniformly random locations and 67 aligned points). The lines shown correspond to the fitted fracture lines that are ranked higher than the rank percentile values of 20%, 40%, 60% and 80%. We have found these threshold values to be useful for detecting major orientations from a complex map of lines. For visual clarity in Figure **4.1**, where there are clusters of lines of similar orientation, the main direction appears as a dense line; the feint lines are neighbouring lines of similar orientation (see α-blending in Unwin et al. 2006). Although the resulting orientations in Figure **4.1** appear promising, the assessment is visual and, therefore, subjective and in practical applications the selection of threshold values instead of quantiles would require some trials.

Note that in three dimensions, triples of points are used to build a candidate plane; these are established sequentially in the same manner as in two-dimension, and the rest of the procedure is unchanged. BFS, in its standard form, for medium to large point clouds, in addition to its high computational time, also requires significant memory (measured in terabytes). However, the memory issue can be efficiently addressed by implementing a fixed storage size similar to the stack structure of, say, $n_{best} = 1{,}000$ rankings, that is, only n_{best} rankings of the best matches are stored.

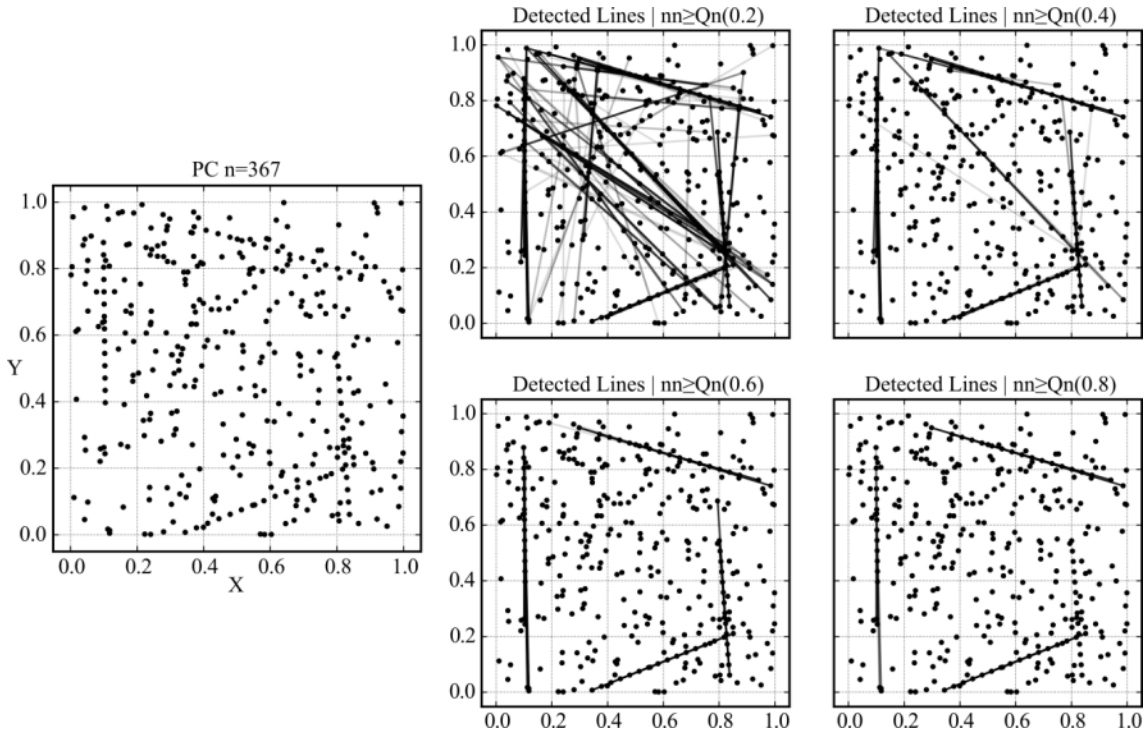


Figure 4.1: A demonstration of BFS applied to a point cloud with a total of 367 points in two dimensions comprising 300 random points superimposed on a set of 5 lines discretised into a total of 67 points. Qn in the titles stands for quantiles computed from the ranks.

A Line Cluster Fitting (LCF) procedure was used to choose automatically the most likely orientations of a set of similarly oriented fracture lines (Figure **4.2**). The proposal removes the need for visual judgment and therefore increases the efficiency of the automation. The method was successfully applied to the orientations resulting from the BFS method to derive the main orientation line fittings as shown in Figure **4.2**. Hereafter we refer to the full procedure as Enhanced BFS. The LCF procedure is as follows.

1. Select sequentially a line $l_i | \, i = 1, n-1$
2. Select another line $l_j | \, j = i + 1$
3. Check if l_j is already clustered; if so go to step 2
4. Check whether both endpoints of l_j are within the tol distance from l_i; if so update the cluster C_i adding l_j; otherwise go to step 2
5. Mark l_j as clustered; go to step 1
6. For each cluster find the best fitting line using least squares

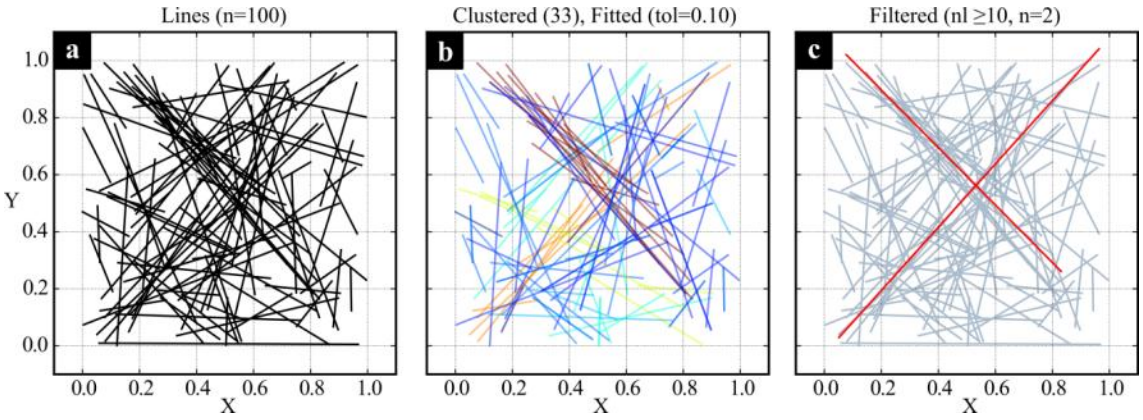


Figure 4.2: LCF in action: Filtering lines and fitting the main orientations using a tolerance=0.1. (a) a small part of the original line data set produced by BFS; (b): clusters of line segments produced by our method and (c) the two fitted main lines (red) as proposed by LCF.

For three-dimensional applications, the fitting process is the same as two-dimension but with three points selected each time to fit a candidate plane. However, the computational cost of the three-dimensional implementation of BFS is significant. This cost is directly related to the number of possible solutions that must be examined,

$$C_{BFS} = N = \binom{n}{k} = \frac{n!}{(n-k)!\,k!}, \tag{4.1}$$

where C_{BFS} (N) is the number of cases to be evaluated. The method can easily become impractical to implement even for a reasonable size point dataset. This combinatorial explosion problem is recognized in the literature. For example, the Habanero dataset contains nearly 25,000 points, for which N=2,603,854,175,000 candidate planes must be constructed and the distances of each plane to the 25,000 points have to be calculated and ranked just to fit one fracture. Memory is also a significant issue unless the technique discussed above is used. Using an iterative fitting process can reduce computation time. In this case, points that have been assigned to fractures are not considered for any further assignment. However, this does not solve the fundamental exhaustive search problem and BFS can only realistically be applied to small data sets ($n < 10k$) depending on the capacity and performance of the computing system used.

4.2.2 RANSAC method

The Random Sampling Consensus (RANSAC) method was originally proposed by Fischler and Bolles (1981) to address the location determination problem (LPD) given an image depicting a set of landmarks with known locations. Although RANSAC originated in image processing it has rapidly found wider application in areas such as robotics, computer vision and, more recently, statistics (Szeliski 2010). Applications include the provision of artificial vision for robotics in surveying or manufacturing, automatic image matching and registration in medicine, environmental science and astronomy (Hartley and Zisserman 2004). The RANSAC method is more robust to contaminating points (outliers) in a point cloud than the traditional least squares methods as typically demonstrated in Figure 4.3.

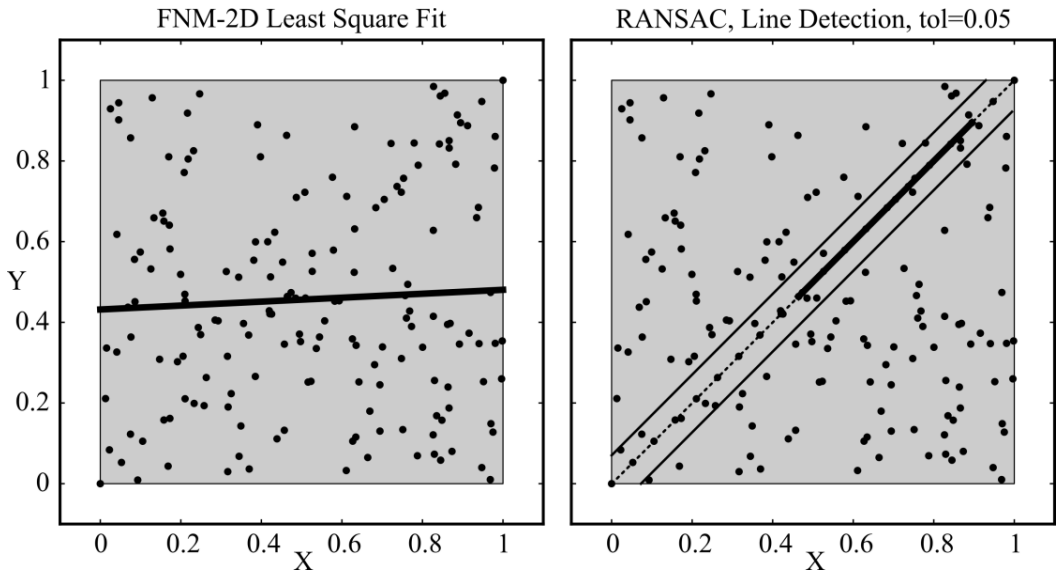


Figure 4.3: Comparison of the performance of Least Squares (LS) and RANSAC for highly contaminated linear objects in a point cloud; RANSAC finds the best fit after 20 iterations which were completed in less than a second. The two lines around the fitted line in the RANSAC result correspond to the tolerance value τ used to compute the cost function.

The RANSAC method is of interest in fracture modelling because of its efficiency in dealing with large point datasets. RANSAC relies on random sampling strategies in which only a selection of cases is examined rather than all possible cases as in BFS. The input and output of the RANSAC method are:

$$\text{inputs}\begin{cases}U=\{x_i\}\\ f(S): S\rightarrow p\end{cases} \quad \text{and} \quad \text{outputs}\begin{cases}C(p,x)\\ p^*\end{cases}, \tag{4.2}$$

where U is the set of points used and $|U|=N$, f is the fracture fitting function (e.g., least squares method) used to compute fitting parameters p given a sample S from U, $C(.)$ is the cost function given a set of data points x and fitting parameters, p and p^* are the final optimized fracture parameters that minimize the cost function. The simplest implementation of the RANSAC algorithm is as follows.

1. Initiate $n_R=0,\ k=0$, $O^*=0$ and $C^*=0$
2. Set $k=k+1$ as the trial counter
3. Select a random set of points of size m $S_k\subset U, |S_k|=m$
4. Compute parameters of the fitted fracture $p_k=f(S_k)$
5. Evaluate the cost function $C_k(p_k,x)$, $\forall x\in U$
6. Update the objective function for trial k that is $O_k=n_k/C_k$ where n_k is the number of inliers
7. Record the best set so far if $O^*<O_k$ then $O^*=O_k$ and $p^*=p_k$
8. Repeat from step 2 until $k>n_p$ where n_p is the number of trials chosen
9. Record the fitted model and remove associated points from U
10. $n_R=n_R+1$ and restart from step 1
11. Finish if all points have been associated or $|U|\leq m$ or $n_R=N$, where N is a number chosen by user

Note that m defines a minimum sample size required for modeling which is 2 or 3 given two- and three-dimensional points for fitting a fracture line or plane respectively. The cost function C_k is defined as the sum of the distances between the points and the candidate fracture model. Inliers are those points located within a tolerance distance τ (i.e., fitting bandwidth) of the fitted model. The objective function O_k can be proposed given the cost function to maximize the number of inliers n_k in fitting process. Indeed for the same values for n_k in trials the cost function proposes the best solution. That is n_k is a critical value for assessment of the fitting given a tolerance value.

$$n_k = \#\left(D_{ij} \le \tau\right) = \sum\left(\mathbf{1}_\tau(D)\right), \quad \mathbf{1}_\tau\left(D_{ij}\right) = \begin{cases} 1 & if\ D_{ij} \le \tau, \\ 0 & if\ D_{ij} > \tau. \end{cases} \tag{4.3}$$

where $\#(.)$ is a count function, D_{ij} is the set of distances between points (j) and the candidate fracture (i), τ is the distance tolerance to determine the point-fracture association and $\mathbf{1}_\tau(.)$ is an indicator function applied as defined. The value of τ depends on the application and the accuracy required.

The number n_p of trials required to find a best solution is defined as follows.

$$n_p = ceil\left(\frac{\log(1-\tilde{p})}{\log(1-(1-q)^m)}\right), \tag{4.4}$$

where q is the probability that any selected data point is an outlier (Fischler and Bolles 1981); $\tilde{p}$ is the probability of success and m is the minimum number (2 and 3 for line and plane, respectively) of points to satisfy the model. Further discussion on evaluation of n_p can be found in Papazov and Burschka (2010). Figure **4.4** provides performance curves for the probability $q \in [0.1,0.3,0.5,0.7,0.9]$ in detecting lines (Figure **4.4**a) and planes (Figure **4.4**b).

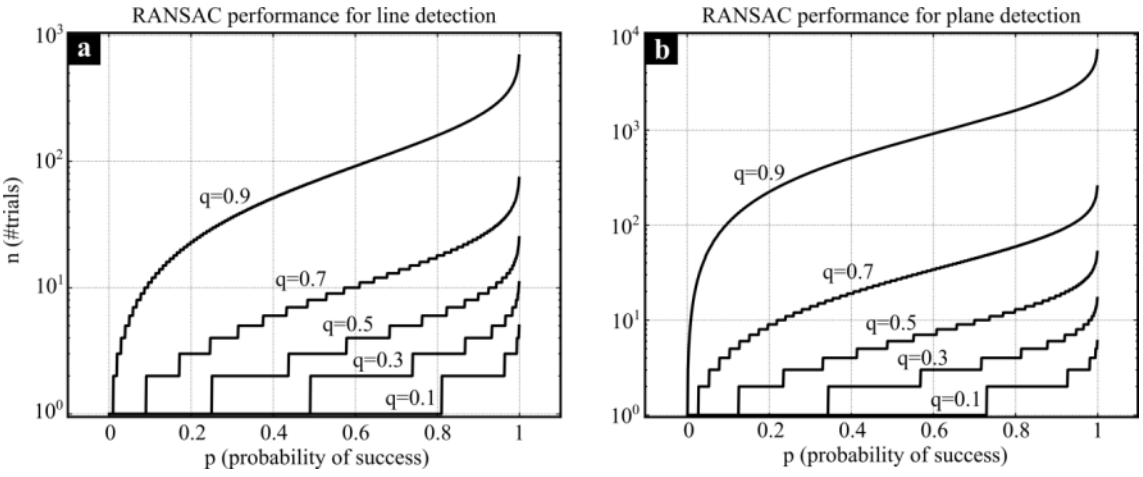


Figure 4.4: Performance of RANSAC for line (a) and plane (b) fitting; q is the probability of outliers.

An important step in the RANSAC implementation is the determination of the association of seismic points with the candidate fracture. In the proposed implementation, the criterion for association is the maximum distance, τ, of points

from the candidate fracture; all points within this distance from a candidate fracture are termed inliers. This definition for point-fracture association is completely general in terms of the geometrical shapes used to represent fractures and, for example, applies equally well to curved fracture surfaces. A detailed discussion on general optimization for the cost function can be found in Yaniv (2010).

Completion of the steps in the RANSAC algorithm given above will yield a fitted fracture, which is the best candidate, defined as the one with the highest rank, among all those examined. The associated seismic points are then removed from the dataset and the process is repeated to fit the next fracture; this process is repeated until there are fewer than 2 or 3 points remaining, for two- and three-dimensional cases respectively. This step is particularly important for RANSAC because: (*i*) the probability of finding the next best fitted fracture within a limited number of trials increases, and (*ii*) the efficiency of objective function evaluation increases with fewer numbers of points.

The computational cost (e.g., elapsed time) of RANSAC is much less than that of BFS and can be estimated as $T_{RANSAC} = n_R.n_p.t_M$ where t_M is the time required to compute the model. We conclude this section by introducing an extension to RANSAC, R-RANSAC, which provides even better performance. In Sect. 4.3, some performance evaluations are conducted using both a simulated and real case data sets.

R-RANSAC

Randomized RANSAC (R-RANSAC) is an extension to increase the efficiency and accuracy of the implementation (Matas and Chum 2004). The idea is simple: under normal conditions the majority of data points are outliers and an improvement could be achieved if the evaluation of those candidate lines/planes for which the cost function is comprised largely of outlier points is done rapidly by means of shortcuts. The cost function increases from a set of points comprised entirely of inliers to a set of points comprised entirely of outliers; the cost function for a random selection of points from a point cloud will tend to be at the higher end.

A quick assessment is sufficient to reject those candidates that lack sufficient neighboring points.

4.2.3 Fracture Extents

The final extent of each detected fracture generated by RANSAC is determined by fitting a least squares line in two-dimension to the points associated with the fracture. In three-dimensions a three-dimensional convex-hull can effectively encapsulate all the points associated with the fracture and thus defines the extent of the fracture. For small values of τ (distance tolerance) the least squares line and the fitted fracture line will be almost identical in two-dimension in terms of the orientation. In three-dimensional cases the associated points are projected onto the fitted fracture plane for the fitting of a two-dimensional convex hull. The projection may cause some changes in the two-dimensional polygonal shape of the fracture compared with the three-dimensional convex-hull fitted without the projection. In our implementation we considered only small τ values as this is the case for practical applications in fitting fractures (the aperture to extent ratio is very low) to a point cloud so the discrepancy is expected to be small.

4.3 Results and Discussions

In this section, the EBFS and RANSAC methods are applied to some simulated datasets and to the Habanero seismic point cloud dataset and their performances are compared. Both two- and three-dimensional cases were examined for detecting fracture trace lines and fracture planes, respectively.

4.3.1 EBFS and RANSAC applied on two-dimensional Point Cloud

A flexible procedure to generate a two-dimensional point cloud in which there is a known number of lines randomly simulated and discretised into a few oriented points is as follows.

1. pc = generate n random pairs of (x, y) uniformly
2. generate k number of two-point lines uniformly
3. set $m = \mathbf{max}(1, \mathbf{length}(line)/d)$

4. for each line generate m points between its endpoints
5. append the point to pc

The number of lines can readily be changed, as can the resolution of discretization and the number of randomly positioned points. Such a point cloud with hidden lines (inliers) among noise (outliers) provides a robust means of assessing the performance of the methods.

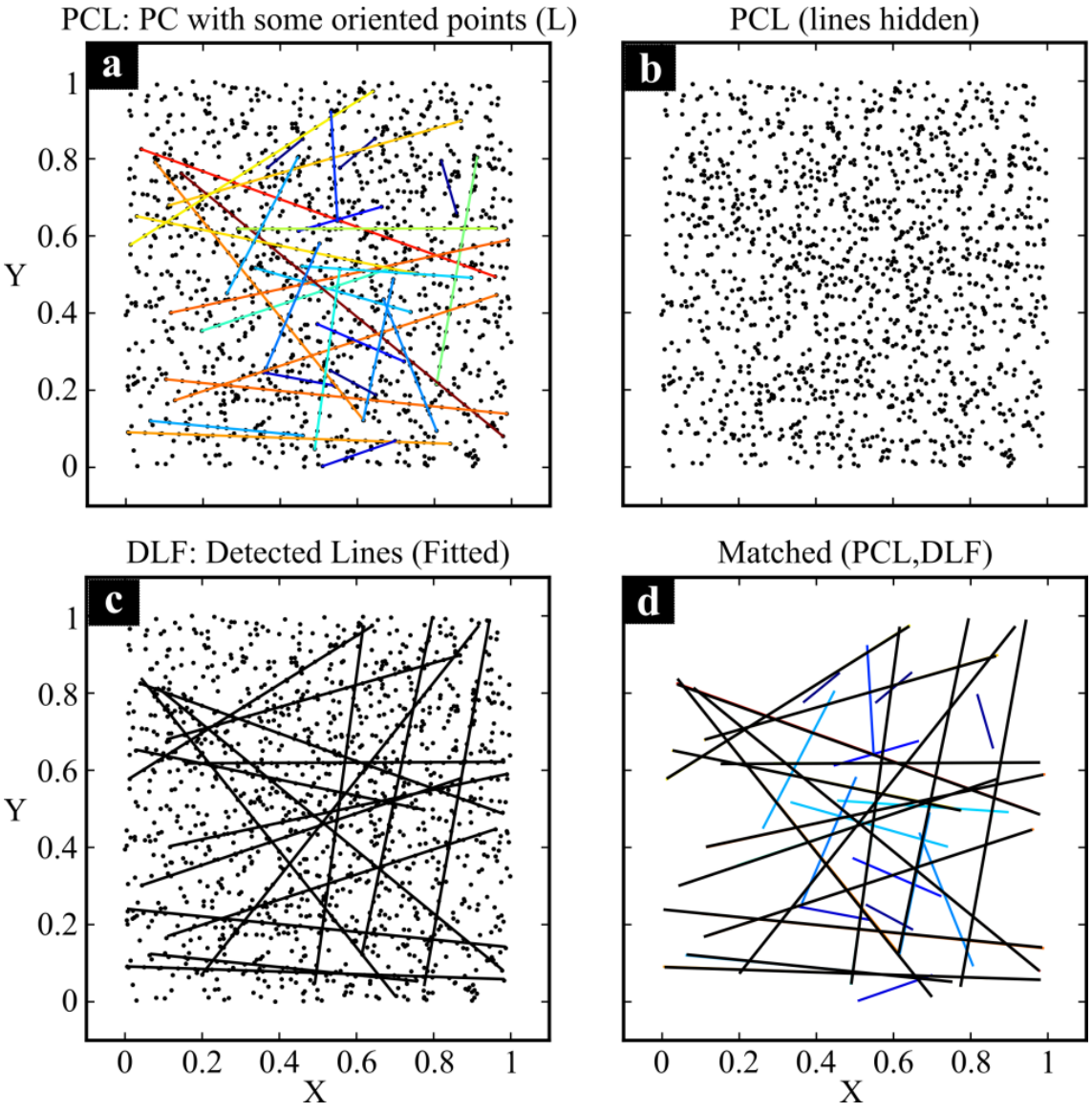


Figure 4.5: Application of EBFS to a two-dimensional point cloud comprising 1,365 points; (a) point cloud with 30 embedded fracture lines represented using 365 points; (b) the same point cloud without the lines drawn; (c) lines detected by EBFS; (d) detected fracture lines in (c) are superimposed on the 30 initial hidden lines in (a).

As demonstrated in Figure **4.5** EBFS performs well on a heavily contaminated dataset of linear objects. In the example given, a total of 30 different fracture lines (shown on Figure **4.5**a as color-coded lines using the blue-red spectrum as the length increases) are represented by a total of only 365 points, which are heavily contaminated by 1,000 randomly distributed points. As seen in Figure **4.5**b, it is almost impossible to detect the 30 linear objects by visual inspection. After applying EBFS and using the fitting condition that candidate lines must have at least a certain number of associated points (here 14, for example) defined by a distance tolerance of 0.01, the final fitted fracture lines are shown in Figure **4.5**c. Clearly the longer linear objects are satisfactorily detected but shorter linear objects are not detectable using the specified parameters. This can be seen easily in Figure **4.5**d in which the fitted fracture lines are superimposed on the original 30 lines.

The procedure comprises two stages: (*i*) find candidate lines and, (*ii*) choose clusters of lines. A summary of the complete algorithm is as follows.

1. $cands = \mathbf{PCL}(pc, tol, cn)$ all candidate lines according to the conditions: points within bandwidth (= tol) and at least cn number of points involved,
2. $c, fits = \mathbf{LCF}(cands, ltol, nl)$ where $ltol$ is the bandwidth to involve adjacent lines, nl is the minimum number of lines, c is the clusters of lines and $fits$ is a collection of fitted main orientation lines.

The RANSAC method was applied to the same simulated data set with a few different setups as described below. First, four different numbers of iterations (50, 100, 200 and 400) were applied to investigate the performance and accuracy of the method. The resulting fracture networks are shown in Figure **4.6**. It is apparent from these networks that increasing the number of iterations improves the accuracy of the method; however, the relationship is not linear. For example, the improvement achieved from 50 and 100 iterations is minimal.

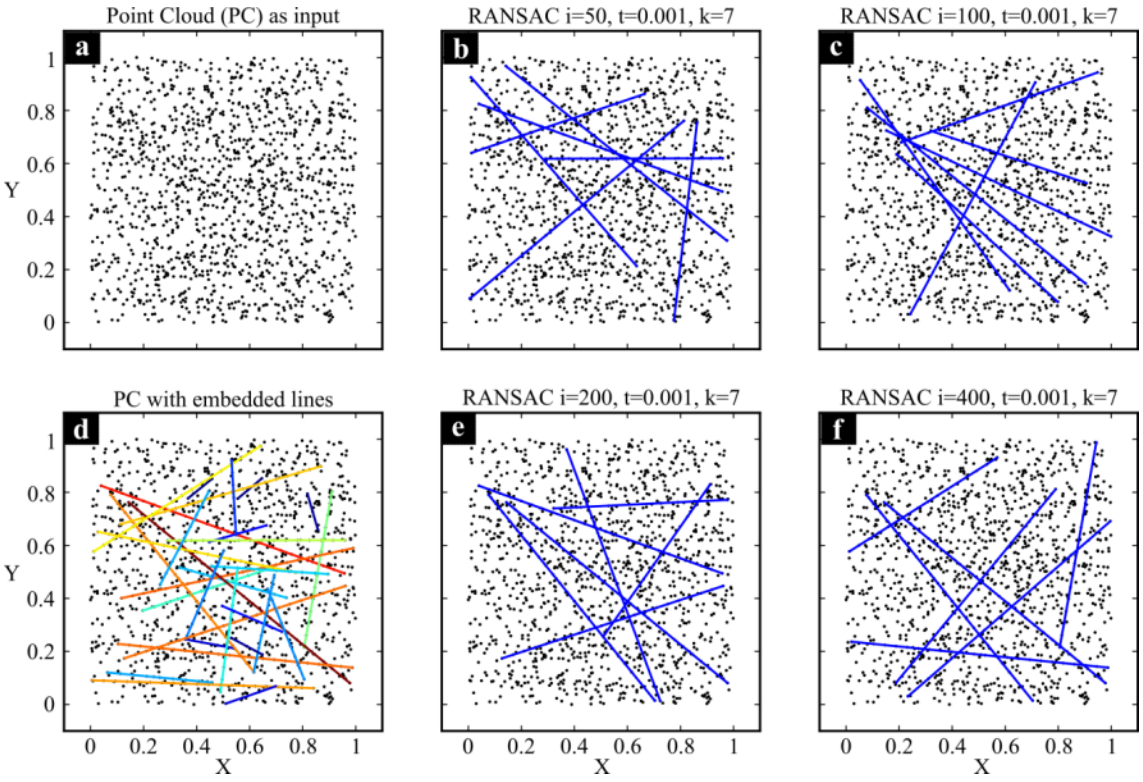


Figure 4.6: Lines resulting from RANSAC for different numbers (i ∈ [50,100,200,400]) of iterations per stage. The embedded lines are shown in (d). As the number of iterations increases the large numbers of linearly aligned points are correctly identified as embedded lines as shown in subplot (f).

In addition, the tolerance value has to be adjusted according to the strength of linearity within the point cloud. For the example given here, the point cloud is highly dispersed and a very small tolerance value was required to achieve acceptable results. In practical applications, the choice of tolerance value will be informed by any available information about the nature and geometry of the fractures. The fracture detection or measurement method (e.g., seismic methods) may also assist in the choice. A diagnostic approach for selecting an appropriate tolerance is to evaluate the final fracture network in terms of the overall setting such as the number of fractures. Figure **4.7** demonstrates the effect of the tolerance on the fitting process.

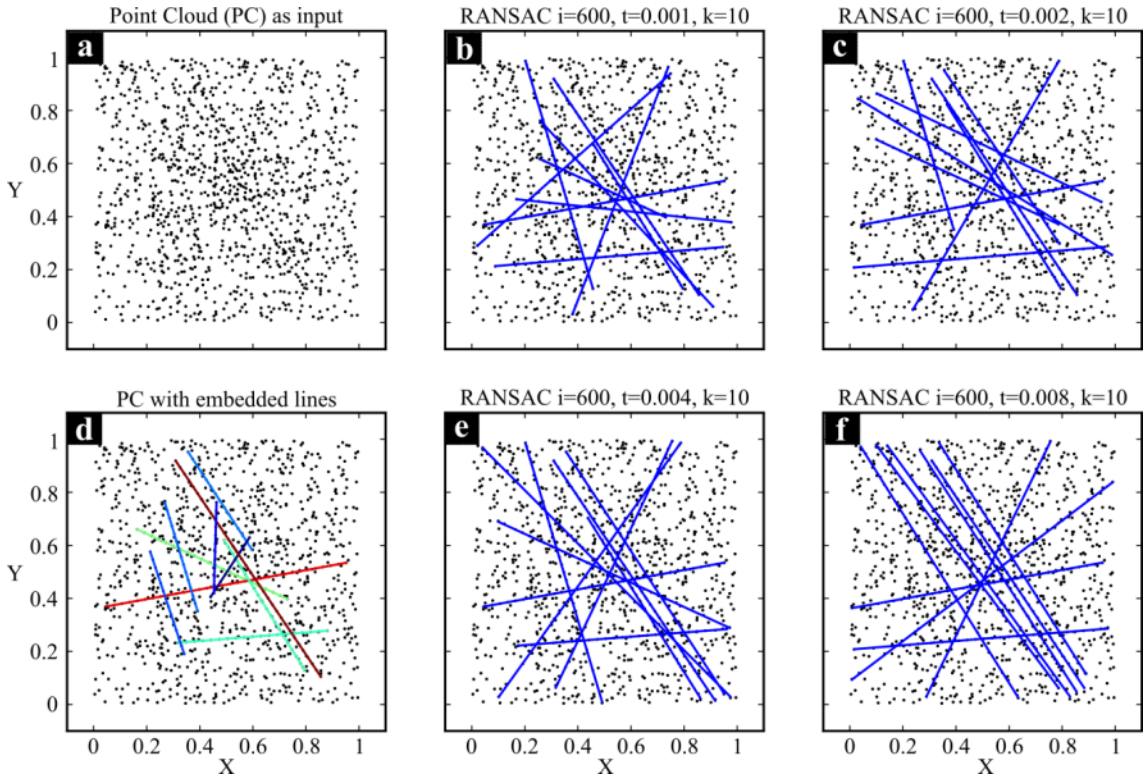


Figure 4.7: A demonstration of the effect on the RANSAC fitting process of varying the tolerance value. Figures (b) and (f) demonstrate clearly the importance of choosing the correct tolerance. Note that in reality, for example, field measurements, the ratio of overall inliers over outliers is much higher than that (0.192) used in these examples.

We assessed the sensitivity of RANSAC to varying the number of trials per stage, n_p, and the distance tolerance, tol. To do so, we used a criterion based on the cosine similarity measure between two lines. We adapted this measure to include the distances between lines as weighting factors: closer lines receive higher weights. Figure **4.8**a shows the results for a simulated point cloud comprising 1,000 random points and 157 points representing ten embedded lines. For small values of n_p the lines detected are less similar to the embedded lines and significant fluctuations may occur in the similarity measure from one value of n_p to another. For values of n_p greater than 200, however, the similarity is consistently high (>0.9) which suggests that, for this example, 200 is an optimum value for the number of iterations per stage. Figure **4.8**b shows the effect of varying the distance tolerance on the similarity between generated and embedded lines; on average, for tolerances greater than 0.01, the similarity decreases as the tolerance increases. Although the average similarity is around 0.9 for $tol < 0.10$ the significant

fluctuations in similarity values makes it difficult to draw general conclusions. A summarized procedure is as follows.

1. Choose l_1 from the lines set 1;
2. For each line l_2 from lines set 2, find the cosine similarity (sim) measure as follows. $similarity = \left|\frac{\mathbf{u}\cdot\mathbf{v}}{|\mathbf{u}|.|\mathbf{v}|}\right|$ where $\mathbf{u}$ and $\mathbf{v}$ are unit vectors of the lines l_1 and l_2;
3. Find distance ($dist$) between centroid points of the lines l_1 and l_2;
4. Find z-score transforms of sim and $dist$ to the range of [0,1];
5. Return the value of $esim = sim * (1 - dist)$ as the new measure of similarity;
6. Repeat from step 1 for all lines in set 1.

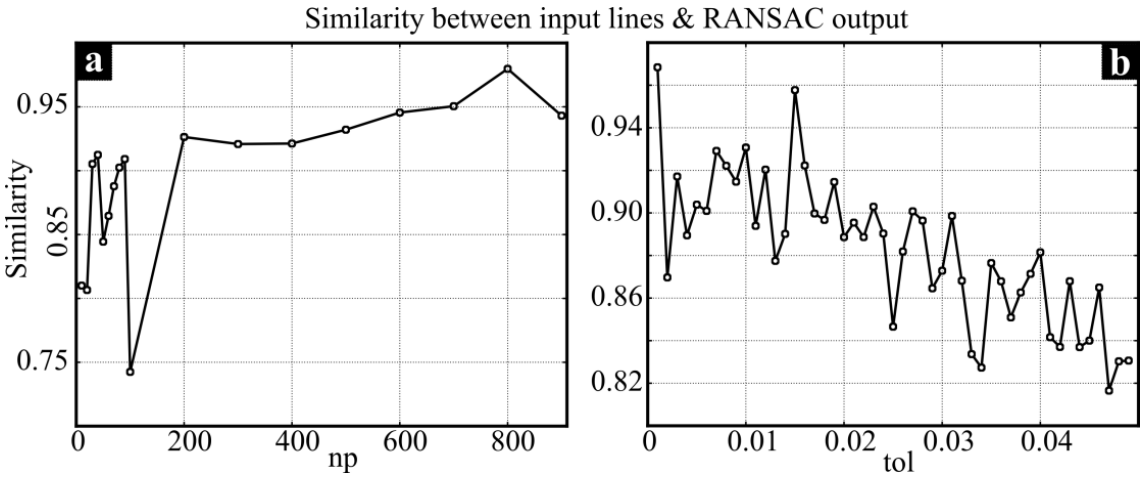


Figure 4.8: Performance of RANSAC as a function of varying the number, n_p, of trials per stage and a constant distance tolerance equal to 0.001 (a) and varying the distance tolerance tol with n_p fixed at 200 (b)

We reiterate a fundamental difference between RANSAC and BFS as implemented here. In RANSAC the points associated with a fitted fracture are removed from the dataset before the procedure proceeds to fit the next fracture. In BFS, points are not removed at any stage of the fitting process and any point could, therefore, be associated with several different fractures.

4.3.2 EBFS and RANSAC applied to a simulated three-dimensional Point Cloud

A simulated point cloud with a total of 609 points was prepared to test the two methods. The point cloud includes three sets of planar points representing three three-dimensional polygonal fracture surfaces. A total of 109 points is used to represent the three fractures and the remaining points were generated in the three-dimensional space using a Poisson process. The high level of contamination ($\frac{500}{109} = 4.6$) provides a challenging case on which to test the performances of both methods. To provide a more realistic example, the locations of the inliers were randomly moved, using values from a uniform distribution, around the associated planes with a maximum of 1% of the dimensions of the volume of study (a cube). Figure **4.9**a shows the point dataset used for the test with the points representing the three fractures shown as open circles.

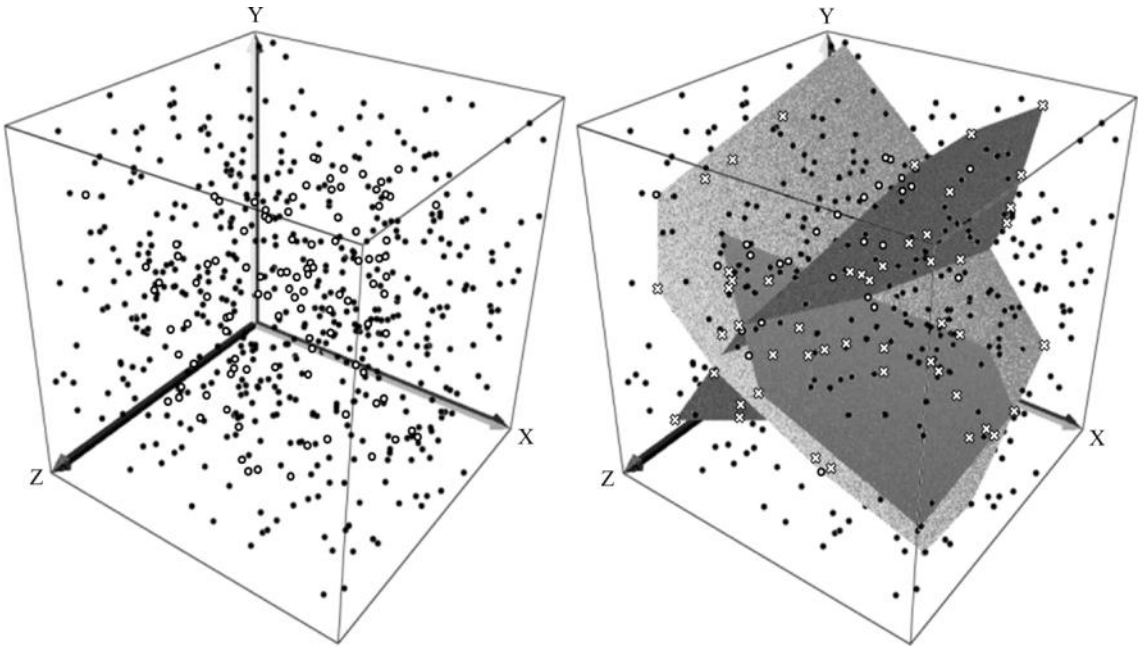


Figure 4.9: Plane detection using EBFS method. (a) three-dimensional point cloud comprising 109 oriented points (open circles) representing three fracture surfaces and 500 points (filled circles) with coordinates from a Poisson process; (b) resulting fractures (only the first three are shown). Crosses are random points associated with the detected fractures.

Figure **4.9**b shows the first three fitted fracture polygons. The resulting orientations include points representing embedded fractures excluding those that

exceed the distance threshold (open circles). In addition, some random points close to the fractures were included in the fitting stage (crosses). Figure **4.10** shows the number of points incorporated in the chosen best matches for the iterations. Note that although EBFS evaluates all possible candidates, the majority of the points are associated with the earliest selected best candidates.

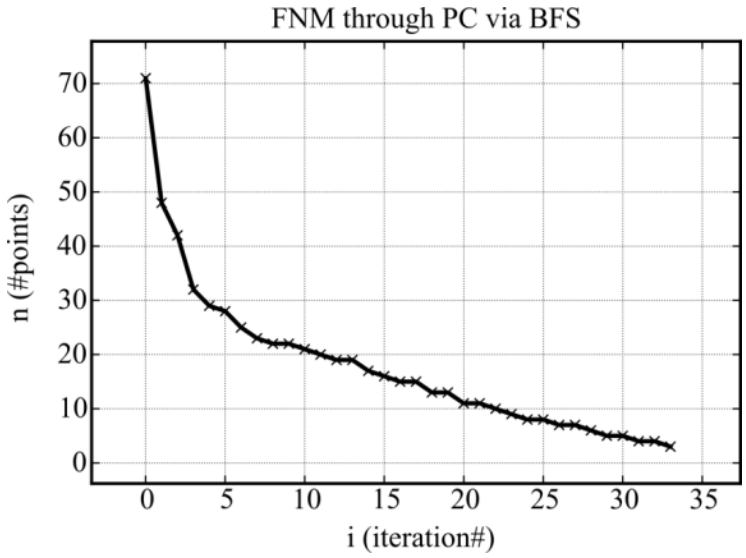


Figure 4.10: The greatest numbers of points are associated with the earliest selected planes.

RANSAC was applied to the same point cloud with parameters: distance tolerance = 0.01, extent $\{(x, y)|x, y \in [0,1)\}$ and number of trials per stage $n_p = 500$. If there is any indication of the orientations in the point cloud a small number of iterations per stage may be sufficient to find the best configuration by inspecting the results; otherwise, depending on the code and the hardware, the number of iterations may be limited by the computing time required for higher repetition strategies. The resulting fitted fractures are shown in Figure **4.11**.

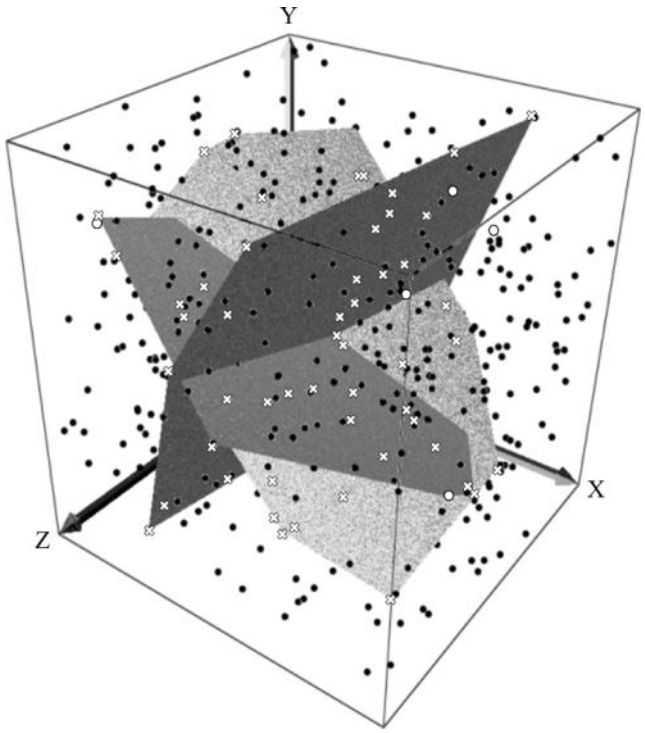


Figure 4.11: The fracture network generated by applying RANSAC to the simulated point cloud. Compared with BFS, RANSAC, with $\{tol = 0.01; mcn = 500\}$ achieves an acceptably accurate result at significantly reduced computational cost, $C_{RANSAC} = C_{BFS}/1208$. The efficiency of RANSAC increases further as the amount of data increases.

4.3.3 The Habanero Seismic Events Point Cloud

The Habanero hot dry rock geothermal project is located in the Cooper Basin of South Australia. The data were collected during fracture stimulation in November and December of 2003. The data are shown in Figure **4.12**.

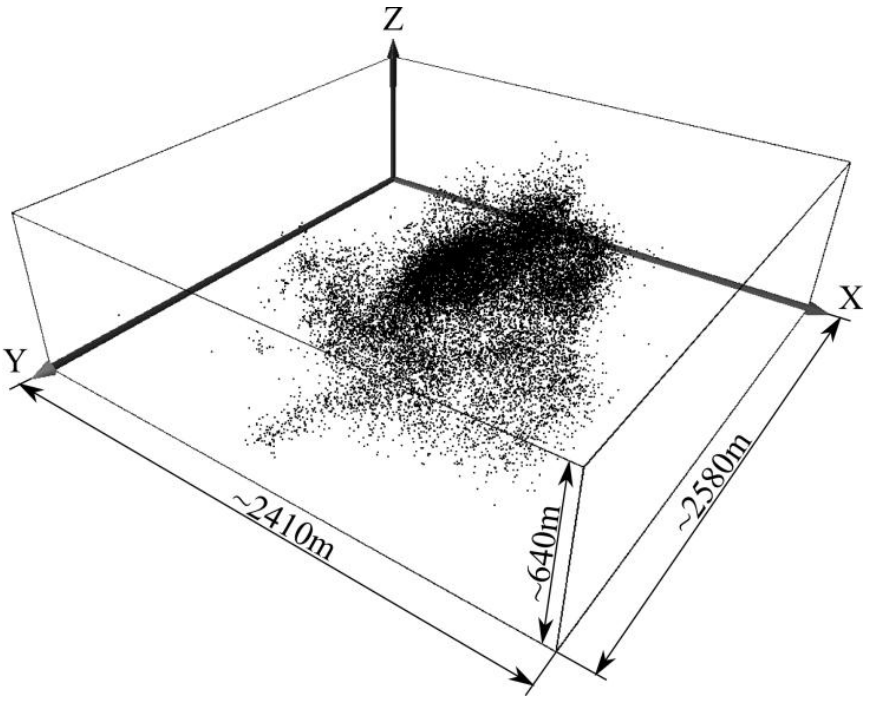


Figure 4.12: Habanero point cloud data corresponding to seismic events recorded in the 2003 fracture stimulation. The number of points is 23,232 and the cloud is approximately horizontally oriented.

BFS is not suitable in this case as it is only practical for small datasets. The following parameters were used for the RANSAC fracture fitting process:

- The coordinates are scaled to [0, 1] with the aspect ratio of 1 preserved.
- Distance tolerance = 0.001.
- Number of trials per stage = 500.
- The fitted fractures are planar convex polygons.

Figure **4.13** demonstrates an exponential relationship for both the time consumed in fitting one fracture in each stage and the number of points per fracture polygon as the RANSAC algorithm proceeds. These relationships prove that the better matches (fracture with a greater number of associated points) are found first, as expected. The graphs are also consistent with the results obtained from BFS shown in Figure **4.10**.

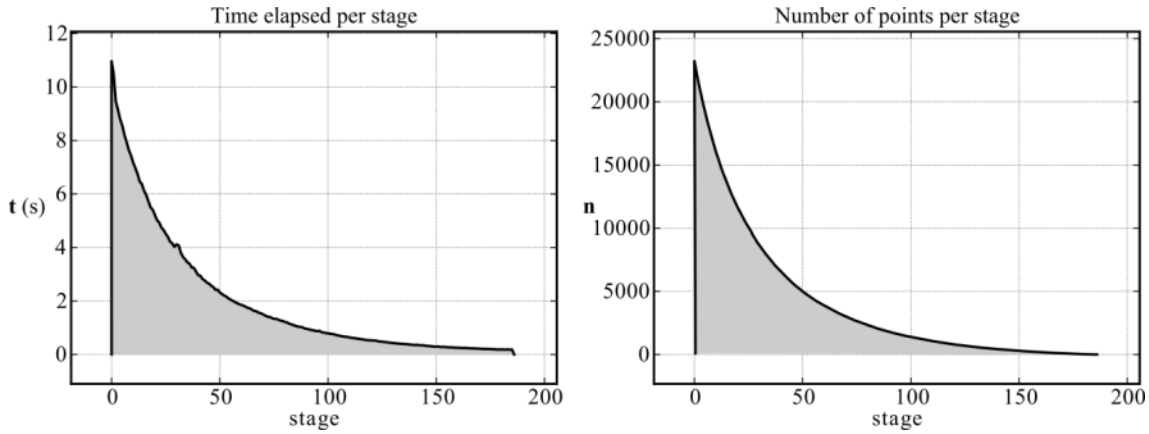


Figure 4.13: The construction of the first polygons requires assessment of the largest number of points and the greatest amount of time with the number and time declining exponentially with the number of stages.

The final results of the fitted fracture model for the Habanero reservoir are given in Figure **4.14**. According to Wyborn et al. (2005) based on extensive analyses of geophysical data it is believed that most of the fractures in the reservoir are approximately horizontal. This feature is clearly supported by the model generated by RANSAC as shown in Figure **4.14**. Figure **4.14**a shows all fitted fracture polygons; Figure **4.14**b shows those fracture polygons dipping at an angle less than or equal to 15 degrees, a total of 143 out of 186 fractures; Figure **4.14**c shows the first three fracture polygons found by the RANSAC algorithm.

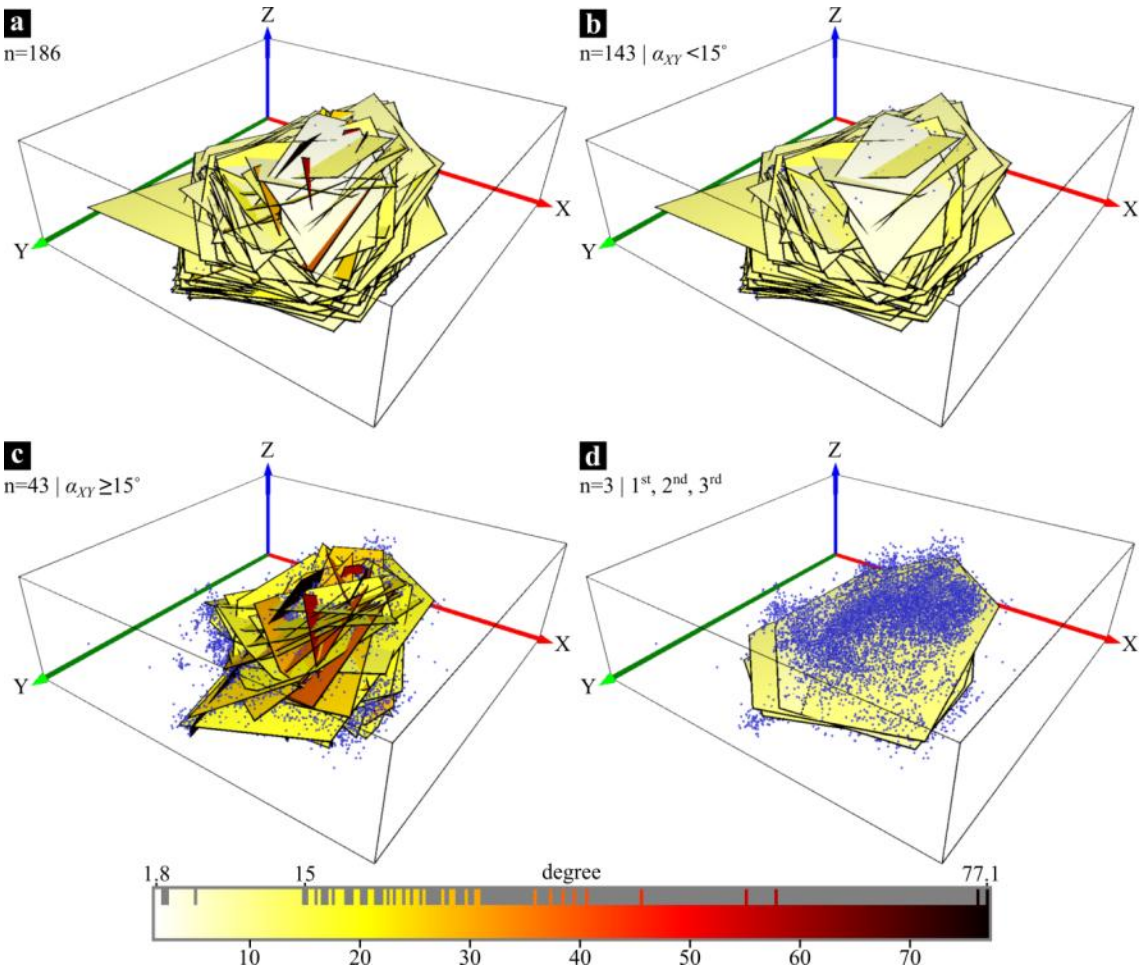


Figure 4.14: Fractures fitted to the Habanero seismic point cloud data; (a) All 186 polygons where the first one involved 956 associated points, the second one 857, the third 786; (b) The 143 fitted polygons that have dip angle less than or equal to 15 degrees; (c) The first three polygons fitted by RANSAC.

We also conducted statistical analyses of the number of points associated with the areas and dip angles of fracture polygons and the number of edges per fracture; these are given in Figure **4.15**. The histogram of fracture-associated points indicates that the majority of points are associated with the first few detected fracture planes. The points associated with each fitted plane form the polygonal shape for each fracture with the number of edges indicating the complexity of the shape. As shown in Figure **4.15**, the complexity decreases as the procedure proceeds. An interesting finding is that the majority of fitted fractures are almost horizontal and these were the first to be fitted. The histogram of the dip angles of the fitted fractures for this data set is given in Figure **4.16** in which the histogram appears almost lognormal with a mode of 10° to 15°. All of these results are

consistent with the published view that the reservoir response is broadly equivalent to an effective large horizontal fracture that may be regional.

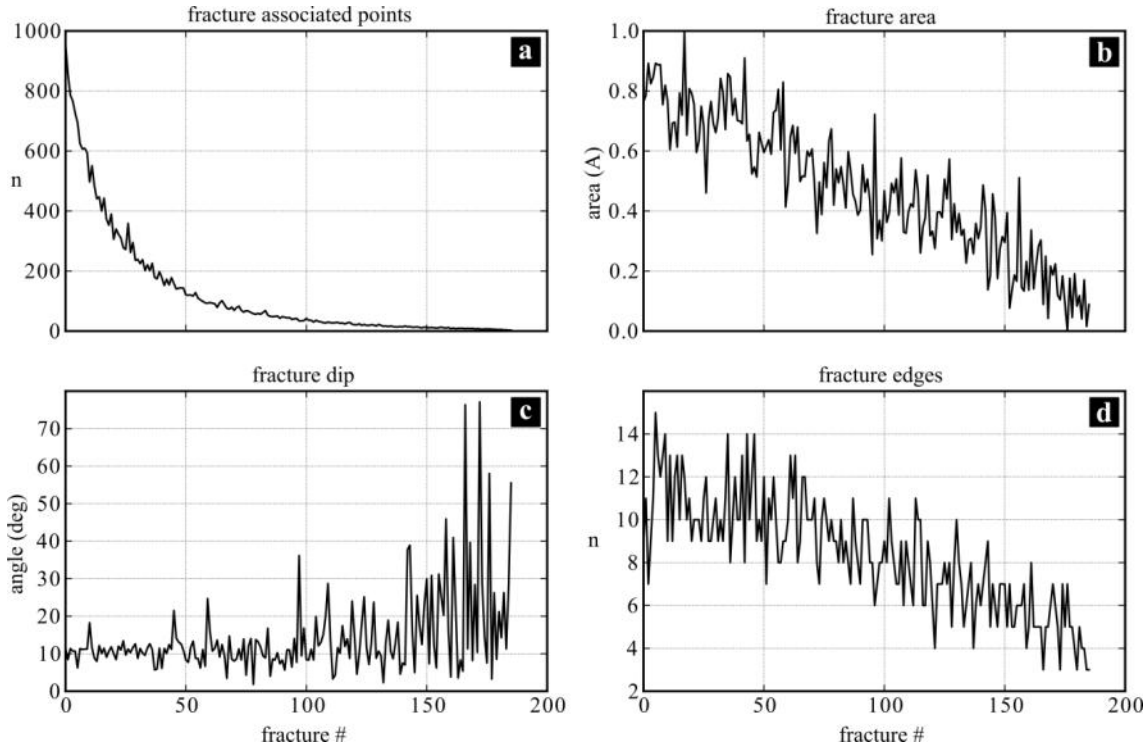


Figure 4.15: (a) The number of points associated with the fitted fractures for the Habanero point cloud; (b) the areas of fractures; (c) the dip angles; and (d) the number of edges of the fitted fractures.

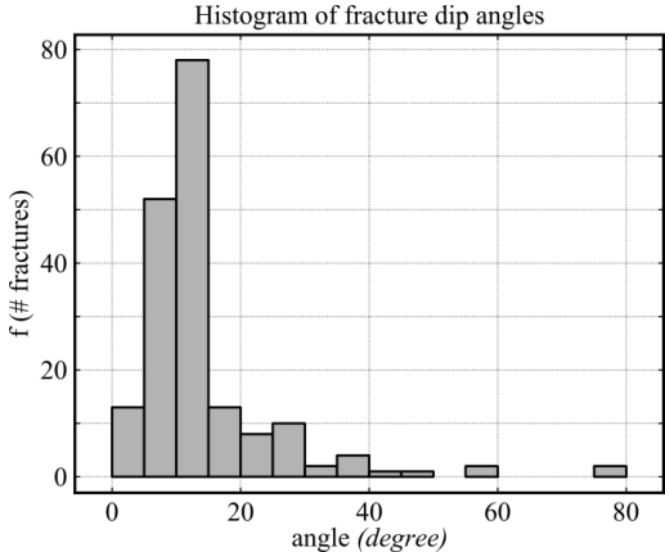


Figure 4.16: Histogram of dip angles for 186 fitted fractures for the Habanero seismic point cloud.

4.4 Conclusions

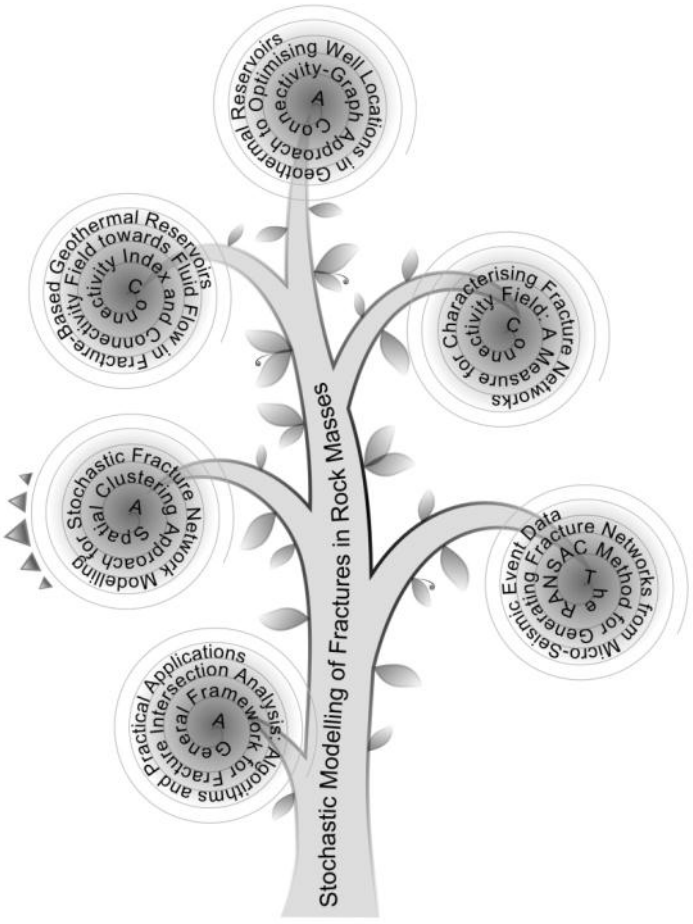
We have adapted the RANSAC method for application to fracture network modelling from point data, in this case from seismic events. We have also proposed an enhanced version of BFS by adding post-processing algorithms in which adaptations have been made to achieve better performance. EBFS has been presented as background for the RANSAC method and as a performance benchmark for it. Whilst EBFS provides an intuitive and straightforward approach to fitting fractures to point cloud data, its use is restricted to relatively small data sets; as the size of the point cloud increases the method rapidly becomes impractical due to the increase in computation, memory and time. RANSAC is a robust fitting method in general and, in particular, has been shown to be an efficient method for fitting fractures to point data, using reasonable computing time and memory. We have discussed the effect on RANSAC performance and output of the choice of model parameters such as the distance tolerance for associating points with a candidate fracture and the number of trials per stage. We have also proposed a set of new assessment approaches with applications to quantify the resulting simulated fracture network in terms of the new similarity measure and the efficiency.

We have presented case studies using simulated point cloud data in two- and three-dimension and a real data set comprising seismic events resulting from a fracture stimulation process conducted in the Habanero geothermal energy system in 2003. In both types of point cloud the resulting fractures are very well matched either with the embedded fractures or with the interpretations in technical reports.

Acknowledgements The authors acknowledge Geodynamics Inc. for providing the Habanero seismic point cloud data set. The work described here was funded by Australian Research Council Discovery Project grant DP110104766.

5

A Spatial Clustering Approach for Stochastic Fracture Network Modelling



The objective of this chapter is to determine a fracture model from micro-seismic conditioning data by minimising the sum of the distances from the data points from the fitted fracture model. The paper documents the use of simulated annealing and a proposed goodness of fit measure. A DD transform is presented and shown to improve the solution both in terms of efficiency and accuracy.

Statement of Authorship

Title of Paper	A Spatial Clustering Approach for Stochastic Fracture Network Modelling.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Seifollahi S, Dowd P.A, Xu C, Fadakar-A Y, 2013, A Spatial Clustering Approach for Stochastic Fracture Network Modelling. Rock Mechanics and Rock Engineering, 10.1007/s00603-013-0456-x.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (~~Candidate~~)	Sattar Seifollahi		
Contribution to the Paper	All concepts, definitions, proposals and the algorithm. More than 95% of the code. Writing the first draft of the paper and all corrections and additions requested by my supervisors.		
Signature		Date	14/3/2014

Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to the development of concepts, co-supervising development of work, manuscript evaluation, significant editing and extensive re-writing of several drafts.		
Signature		Date	14/3/2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to the development of concepts, co-supervising development of work, manuscript evaluation and significant editing and re-writing of the initial draft.		
Signature		Date	14/3/2014

Name of Co-Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	Assistance in developing the proposals and concepts. Some coding for simulations and data analysis and for preparation of some figures. Assistance with an early draft of the paper.		
Signature		Date	18 03 2014

A Spatial Clustering Approach for Stochastic Fracture Network Modelling

S. Seifollahi, P. A. Dowd, C. Xu, A. Y. Fadakar

Abstract Fracture network modelling plays an important role in many application areas in which the behaviour of a rock mass is of interest. These areas include mining, civil, petroleum, water and environmental engineering and geothermal systems modelling. The aim is to model the fractured rock to assess fluid flow or the stability of rock blocks. One important step in fracture network modelling is to estimate the number of fractures and the properties of individual fractures such as their size and orientation. Due to the lack of data and the complexity of the problem, there are significant uncertainties associated with fracture network modelling in practice. Our primary interest is the modelling of fracture networks in geothermal systems and, in this paper, we propose a general stochastic approach to fracture network modelling for this application. We focus on using the seismic point cloud detected during the fracture stimulation of a hot dry rock reservoir to create an enhanced geothermal system; these seismic points are the conditioning data in the modelling process. The seismic points can be used to estimate the geographical extent of the reservoir, the amount of fracturing and the detailed geometries of fractures within the reservoir. The objective is to determine a fracture model from the conditioning data by minimizing the sum of the distances of the points from the fitted fracture model. Fractures are represented as line segments connecting two points in two-dimensional applications or as ellipses in three-dimensional (3D) cases. The novelty of our model is twofold: (1) it comprises a comprehensive fracture modification scheme based on simulated annealing and (2) it introduces new spatial approaches, a goodness-of-fit measure for the fitted fracture model, a measure for fracture similarity and a clustering technique for proposing a locally optimal solution for fracture parameters. We use a simulated dataset to demonstrate the application of the proposed approach followed by a real 3D case study of the Habanero reservoir in the Cooper Basin, Australia.

Keywords Global optimization, Simulated annealing, Spatial clustering, Stochastic fracture network modelling, Conditional simulation.

5.1 Introduction

Fracture network modelling, particularly discrete fracture network (DFN) modelling, is critical for the design, evaluation and development of natural energy systems and resources, particularly for those located at significant depth beneath the Earth's surface. For example, the evaluation of fluid flow through a fractured rock mass for water, petroleum and geothermal applications can only be conducted on a proper rock fracture model. Given the nature of the problem, it is in most cases impossible to observe or measure fractures directly on any scale relevant to the problem. Studies are generally limited to sparse, small-scale observations (e.g. on drill cores) or indirect measures such as those provided by geophysical surveys or, in the case of engineered geothermal systems (EGS), micro-seismic events generated during fracture stimulation. There are significant uncertainties associated with indirect measures of variables such as fractures, and in such circumstances the only feasible approach is a stochastic one. Important parameters in DFN are those that define fracture geometries such as size and orientation of individual fractures.

Existing DFN approaches use various statistical and geostatistical techniques; see for example Chilès and Delfiner (1999), Dershowitz and Einstein (1988), Dowd et al. (2007), Lantuejoul (2002), Kulatilake et al. (1993), Lee et al. (1990), Meyer and Einstein (2002), Oda et al. (1987), Rawnsley and Wei (2001), Stoyan et al. (1995) and Xu et al. (2007). Some of the well-known approaches include object-based and grid-block methods and their extensions. Approaches using geostatistics are applied within the context of Poisson modelling (Gringarten 1997; Billaux et al. 1989; Wen and Sinding-Larsen 1997). However, there are difficulties with all of these methods in constructing a realistic fracture network when the only available conditioning data are in the form of seismic events. The seismic point cloud can be used to estimate the geographical extent of the hot dry rock (HDR) reservoir, the amount of fracturing and the fracture geometries (Baisch et al. 2006; Eisner et al. 2010; Sausse et al. 2010; Xu et al. 2013a). In these stimulation processes, it is

reasonable to assume that seismic events occur only on fracture surfaces (Xu et al. 2013b), which is an assumption in the proposed modelling method. However, the recorded seismic events are relatively sparse and, for a given fracture, usually only a few seismic points are recorded. Establishing the fracture network reservoir model conditioned by this seismic point cloud is critical in creating a more realistic and reliable fracture model for HDR EGS. Recently published work includes DFN optimization conditioned by two- and three-dimensional data (2D and 3D) (Mardia et al. 2007a, b; Xu et al. 2013b). In general, the DFN solution in these approaches is delivered via a stochastic optimization approach. Mardia et al. (2007b) use a Markov chain Monte Carlo (MCMC) approach to condition the fracture model to borehole data. An extension of MCMC to include the seismic point cloud in the conditioning of the fracture model was used in Xu et al. (2013b) for a case study from the Habanero well-field. In the former work, the number of fractures and the parameters of individual fractures are modified during the optimization process. However, the application of the method is limited to borehole intersections. Xu et al. (2013b) fix the number of fractures in advance and only the unknown parameters of fractures, such as their size and orientation, are optimized during the MCMC simulation. In the present work, we address the issues of optimizing the number of fractures and conditioning the model on seismic data by means of a general stochastic model in which the objective function is structured to allow the optimization of the proposed goodness-of-fit measure and the sizes of fractures in the model. We use a comprehensive model modification scheme which includes eight proposals, four for updating the parameters of individual fractures and four for adjusting the size of the fracture network. To adjust the network size, two proposals, termed “Split” and “Special-Split”, are used for growing the fracture network and the other two, termed “Joint” and “Special-Joint”, are used for pruning the fracture network. The purpose of the split proposals is to grow the number of fractures in the model by proposing new fractures. The two join proposals are used to reduce the number of fractures in the system by joining two fractures into a single new fracture or by removing a redundant fracture from the network. A new similarity measure is introduced to find the two most similar fractures in a joint proposal. We use a new spatial clustering method, the distance-directional transformation (DD-transform), to determine the parameters of the new

fractures. The DD-transform is based on distances and orientations between pairs of points. This approach increases the efficiency of the optimization by generating a locally optimal solution for the parameters and reduces the number of trials required to reach an optimal solution. The structure of the proposed method is similar to the MCMC approach of Mardia et al. (2007b), in that both use similar proposals for updating parameters of the fractures and for updating the size of the fracture network. However in our approach, simulated annealing (SA) is used to optimize the fitted model. In addition, the proposed method uses a different objective function based on a new goodness-of-fit measure. Our approach is thus a combination of the MCMC proposal and SA.

The performance of the proposed method is demonstrated using a 2D simulated dataset (Figure **5.2**a) and a real case study of Geodynamics' Habanero reservoir in the Cooper Basin of South Australia (Baisch et al. 2006) (Figure **5.6**a).

5.2 The Objective Function

For simplicity, details of the proposed method are described within the context of 2D applications. The extension to 3D is described in Sect. 5.10. In 2D DFN a fracture is represented as a line segment connecting two points in the region:

$$H \equiv [P, Q] \tag{5.1}$$

where $P = (x_1, y_1)$, $Q = (x_2, y_2)$ are the endpoints of H. We can also represent a fracture using four parameters, $w = (w_1, w_2, w_3, w_4)$, where (w_1, w_2) are the coordinates of the centre and w_3 and w_4 are, respectively, the orientation and the length of the fracture.

The sum of the distances of the points (seismic events) from the fitted fracture model has been used as a goodness-of-fit measure (Xu et al. 2013b). In DFN, this measure is defined as the sum of squared distances of events from their nearest fractures as follows:

$$f_d(w) \equiv \sum_{j=1}^{m} d_{jk^*}, \qquad k^* = \underset{k}{\operatorname{argmin}}(d_{jk}) \tag{5.2}$$

where m is the number of points, k the index of the fracture in the network, d_{jk} the squared distance of the jth point to the kth fracture and d_{jk^*} the smallest of all projection distances from the point to all fractures.

However, in applications where spatial characteristics are important, such as in DFN, it is not sufficient to use this measure to quantify the goodness of fit of the model. For example, this measure does not include fracture size, as the point is projected to the nearest fracture regardless of the sizes of fractures. Therefore, DFN using Eq. (5.2) tends to produce fractures that are too large. To ensure more realistic models, we have developed a new spatial distance based measure. The new measure, which includes both distance and orientation characteristics between the data points and the endpoints of fractures, is defined as:

$$d_{jk} = d_1\theta_1 + d_2\theta_2 \tag{5.3}$$

where d_1 and d_2 are the absolute distances between an event and the endpoints of a given fracture and θ_1 and θ_2 are the corresponding acute angles between the fracture and the line segment connected by the event and the endpoints. As the value of the measure is always positive, it can be used in conjunction with the orthogonal distance in Eq. (5.2) to define the total distances of all points to the fitted fracture model. The new measure has two important features: whilst minimizing the distances of the points from fractures, it also minimizes the fracture sizes.

If the orthogonal projection of a point does not intersect a fracture, it is deemed to be an outlier for that fracture and a penalty, based on the nearest endpoint of the fracture to the point, is assigned to it:

$$d_{jk} = \lambda_p \times \min\Big(d(p_j, P), d(pj, Q)\Big) \tag{5.4}$$

where p_j is an outlier, P and Q are the endpoints of a fracture and d is the absolute distance. λ_p is the weight assigned to the outlier, which is set to the maximum possible size of fractures in the network.

5.3 The Simulated Annealing Method

The simulated annealing method derives its name from the physical process of annealing or cooling of heated metals. Kirkpatrick et al. (1983) trace this procedure back to Metropolis (Metropolis et al. 1953), who originally attempted to simulate the behaviour of an ensemble of atoms in equilibrium at a given temperature. Metropolis constructed a mathematical model of the behaviour of such a system that contained a method for minimizing the total energy of the system.

SA comprises two main iterations: outer and inner. In the outer iteration, the temperature T (notional for non-thermodynamics problems) is updated. To do so, we take any initial value T_0 and a reduction ratio $0 < r < 1$ and use the temperature schedule $T_{k+1} = rT_k,\ k = 0,1,2,\ldots$. In the inner iteration, we update the solution. To generate a new solution in the inner iteration, we randomly generate a new solution (or new state) and a uniformly distributed random number u from $(0, 1)$. The acceptance probability is calculated as:

$$A_p = \exp\left(\frac{f^{old}(w) - f^{new}(w)}{T}\right). \tag{5.5}$$

If $u \leq A_p$, the new state is accepted, otherwise the inner iteration is repeated. f^{old} is the best objective function value obtained in previous iterations and f^{new} is the objective function value based on the proposed new state. The state refers to any perturbation in parameters of a fracture or any proposals, as discussed in Sects. 5.5. SA is an efficient stochastic method for finding a global optimum solution of a nonlinear problem. An alternative to SA, applicable in DFN, is MCMC; see Mardia et al. (2007b) and Xu et al. (2013b).

5.4 A Spatial Clustering Technique

In this section, we propose a spatial clustering approach to generate a locally optimal solution for fracture parameters. The major challenge here is to transform the coordinates of seismic points into a new space of distances and orientations so that features in the transformed space are easy to analyse and the required information can be easily extracted. For a given set of points, the aim is to find the midpoints and orientations of the best lines fitted to the points, among all possible candidates with the most number of events co-aligned with the directions. For each event, a fixed number of cells are created, corresponding to equal angle intervals in the interval $-90°$ to $+90°$. The value assigned to each cell represents the number of events aligned with the selected event in the corresponding direction. More precisely, the angle between a line segment (constructed by connecting a candidate event with another event from the dataset) and the horizontal axis is calculated and the count is accumulated in the corresponding cell. The cell with the maximum value corresponds to the orientation of an optimal line passing through that particular point.

For a given set of points an array of accumulated values is obtained, in which the (j,k)-element of the array (j and k) correspond to the jth point and the index of the kth angle, respectively) is expressed as:

$$A_{jk} = \sum_{i \in I} \min\left(v_{ji}(k), \bar{v}\right) \tag{5.6}$$

where $I = \{i \text{ angle of } [p_i, p_j] \text{from the horizontal axis} \equiv k\}$. v_{ji} is the weight, equal to the inverse distance between the points p_j and p_i:

$$v_{ji}(k) = d\left(p_j, p_i\right)^{-\zeta} \tag{5.7}$$

where ζ is a positive real number that reflects the importance of the inverse distances in calculating the counts in Eq. (5.6); we set $\zeta = 0.5$ in our experiments. The weight $v_{ji}(k)$ decreases as the distance between points p_j and p_i increases. $\bar{v}$ is

a positive real fixed number, depending on the inverse distances between pairs of points; to prevent the undue effects of small distances in calculating weights, we set it to five in our simulated experiment. For a set of points, D, algorithm 1 finds the orientations of the best lines through the points.

Algorithm 1: DD-transform

1. Set $A = 0$ where A is an $m \times \bar{k}$ matrix, m is the total number of points and $\bar{k}$ is a positive integer number corresponding to the number of angles (e.g. $\bar{k} = 80$ for angles between $-\pi/2$ and $\pi/2$ with 1° intervals).
2. Select a point, $p_j \in D$.
3. Select $p_i \in D, i \neq j$, and calculate the distance between points p_j and p_i, i.e., $d(p_j, p_i)$.
4. Find the angle between the line segment $[p_j, p_i]$ and the horizontal axis.
5. Update the element (j, k) of the matrix A as follows: $A_{jk} = A_{jk} + min(v_{ji}(k), \bar{v})$, where k is the angle index, j and i refer to points p_j and p_i and $\bar{v}$ is an upper bound for $v_{ji}(k)$.
6. Set $i = i + 1$. If $i \leq m$, repeat from step 3; otherwise go to step 7.
7. Find the maximum value of each row of A (i.e. among $\bar{k}$ elements) and assign the value to vector A_{max}, and also store the angles corresponding to A_{max}, in I_{max}. Set $j = j + 1$. If $j \leq m$, repeat from step 2; otherwise go to the next step.
8. Sort vector A_{max} in descending order and label it $\bar{A}_{max}$. Change the order of rows of I_{max} and D on the basis of the order of elements in $\bar{A}_{max}$ and denote them by I_A and D_A, respectively.
9. Select the first number of I_A and D_A as the orientation and midpoint of the first proposed line.
10. Select the next number of I_A and D_A. If the selected elements are co-aligned with a line (given a threshold depending on the problem domain and the proposal) which was created in previous iterations, go to step 12; otherwise, go to the next step.
11. Define a new line passing through the selected point with the associated orientation.

12. Use a threshold to determine whether each point belongs to at least to one fracture. If so, terminate the algorithm; otherwise, repeat from step 10.

The elements of I_A and D_A are chosen in such a way that the proposed lines go through all points and no duplicate lines are generated. The algorithm is repeated until all lines are detected. If each point is associated with at least one fracture (i.e. lies within a bandwidth of a fracture), the algorithm terminates. As the size of the bandwidth decreases, more lines will be detected and vice versa. The algorithm is used as a supervised tool to increase the efficiency of the optimization. In our model, when required, parameters of one or two "best" fractures are drawn using the DD-transform.

5.5 Proposed Stochastic Fracture Network Modelling

The proposed modelling method incorporates eight proposals. Four proposals are for updating parameters of existing fractures and the rest are for adjusting the number of fractures. The modelling process consists of three phases. After the initialization phase, the parameters of a randomly selected set of fractures $\{H^j\}_{j=1}^{n_s}$ are updated sequentially using SA. The algorithm then optimizes the size of the fracture network by proposing new fractures or removing redundant ones. After each proposal, the stopping criteria are checked. The algorithm terminates when a pre-specified number of iterations have been completed or the temperature parameter has been reduced to a specified threshold value.

5.5.1 Phase 1: Initialization

At the beginning, n initial fractures are set, where n is an arbitrary small positive integer number. To find the parameters for the initial fractures, we choose n points at random from the problem domain as midpoints of initial fractures. The length and orientation of each fracture are drawn from specified probability distribution functions, here lognormal with parameters informed by expert knowledge and standard Gaussian, respectively. Orientations obtained from the Gaussian are truncated at $-\pi/2$ and $\pi/2$.

5.5.2 Phase 2: Updating Parameters of Individual Fractures

Four proposals are used to optimize the fracture parameters. First, a set of fractures is chosen at random from the network. For each fracture, its orientation and midpoint are perturbed sequentially by values drawn from a Gaussian distribution as follows:

$$w_i = w_i + N(0, \sigma^2), \qquad i = 1,2,3 \tag{5.8}$$

where w_i stands for the coordinates of the midpoint or the orientation of the fracture before the perturbation and N is the Gaussian distribution with mean 0 and variance σ^2. For fracture length, we propose a different approach: another fracture, different from the candidate fracture, is chosen at random from the network, and their lengths are exchanged. As no new values of the variable are involved, the distribution of the orientation remains unchanged after this process. It is noted that random sampling can be used as an alternative to the exchange proposal. For each proposal, the objective function is recalculated using the new point associations due to the perturbation. It is now necessary to decide, on the basis of Eq. (5.5), whether to accept the proposed move.

5.5.3 Phase 3: Updating the Size of the Fracture Network

Four proposals are used to adjust the size of the fracture network. Different criteria are used to select candidate fractures in each proposal. For example, the candidate fracture for the split proposal should be a fracture that has a high potential for being split into two new fractures with two different sets of parameters. The potential is assessed by examining the farthest point to the fracture (the longest distance) and applying the DD-transform to the associated points to check for two distinct best orientations.

Once the candidate fractures are selected, the next step is to propose the parameters for the new fractures to which the proposals are to be applied. The DD-transform is used to suggest the orientations and the coordinates of the midpoints of new fractures. Fracture length is generated from the corresponding distribution

function. The final step is to determine, on the basis of Eq. (5.5), whether to accept the proposed new fractures. The steps of the algorithm are summarized as follows.

Algorithm 2

1. Initialize n fractures, $\{H^i\}_{i=1}^{n}$, where n is a random integer number (a small portion of the points).
2. Select three subsets of fractures at random, i.e. $H_1 = \{H^j\}_{j=1}^{s_1}$, $H_2 = \{H^j\}_{j=1}^{s_2}$ and $H_3 = \{H^j\}_{j=1}^{s_3}$, where s_1, s_2 and s_3 are random integer numbers (portions of existing fractures).
3. Update midpoints of fractures in H_1 using SA. If SC are met, terminate the algorithm.
4. Update orientations of fractures in H_2 using SA. If
5. SC are met, terminate the algorithm.
6. Exchange the lengths of pairs of fractures $H^j, H^{j'} \in H_3$. If SC are met, terminate the algorithm.
7. Apply special-split proposal. If SC are met, terminate the algorithm.
8. Apply split proposal. If SC are met, terminate the algorithm.
9. Apply special-joint proposal. If SC are met, terminate the algorithm.
10. Apply joint proposal. If SC are met, terminate the algorithm.
11. Repeat from step 2.

5.6 Optimal Number of Fractures

Finding an optimal number of fractures is important in DFN and for this purpose a growing algorithm can be used. In this algorithm, a few fractures are initially fitted to the points. During the optimization process, the network is allowed to grow, either by inserting a new fracture or by splitting an existing fracture into two new fractures. This process will continue until an adequate number of fractures is obtained, which is assessed by examining the objective function value and its statistics. Another way is by pruning the network, starting with a large number of fractures. There are two alternatives for reducing the network size: removing a redundant fracture or joining two existing fractures into a single fracture. Both the

growing and pruning techniques start with a set of initial fractures and their final solution for the optimal number of fractures should be similar. These two techniques can be integrated into a model so that the network starts with an arbitrary number of initial fractures and the number of fractures varies during the optimization process (an increase or decrease) as given in algorithm 2.

5.7 Growing the Network ("Split" and "Special-Split")

When the number of fractures is not sufficient to give an acceptable fit to the entire dataset (e.g. existence of some points outside a bandwidth of any fracture given a pre-specified threshold), splitting a fracture into two fractures may improve the goodness of fit. It is then important to choose an appropriate candidate fracture and the set of parameters for the two new fractures. A candidate fracture is chosen on the basis of the existence of an isolated point (the largest distance to the fracture) and the potential for splitting into two noticeably different fractures (orientations), which are assessed by applying the DD-transform on the associated points. More precisely, if the proposed two orientations are significantly different, two new fractures are constructed from the midpoints and the orientations. The lengths of the two new fractures are drawn from the fracture size distribution. The steps of the proposal are:

1. For each fracture, find the largest distance from its farthest associated point. Sort the fractures in descending order of their largest distances and select the first S fractures as initial candidates.
2. For each initial candidate, apply the DD-transform to the associated points to determine the two "best" lines. If the difference in the orientations of the lines obtained in step 2 is not significant (\#) for a given fracture, that fracture is discarded from further consideration.
3. Choose a fracture based on the highest distance as the final candidate fracture.
4. Accept or reject the proposal on the basis of Eq. (5.5).

S is an integer number (a small portion of existing fractures). The value of S depends on the number of existing fractures; we set $S = 5$ in our implementation.

Threshold ϑ is set to a small integer number, e.g. 5(°), so as to discard new generated lines with similar parameters. In other words, if the difference in orientations of the two new fractures is less than 5° the initial candidate fracture is removed from further consideration. The coordinates of midpoints and orientations of proposed new fractures are obtained in step 2 by the DD-transform and the length is obtained by sampling from the fracture length distribution. The key difference between the "Special-Split" and "Split" is in step 2 where for the "Split", two new fractures are generated and the old one is removed; while for the "Special-Split", only one new fracture is generated and the original fracture is kept as the second fracture. The need for two different proposals is because the candidate fracture for splitting may itself already be a good fit to some of the associated points. Thus, in step 2, two versions are assessed depending on the importance of the candidate fracture in the fitted model. The "Special-Split" does not require determination of two different orientations and the first element of I_A and D_A obtained by the DD-transform corresponds to the only fracture required in this proposal.

5.8 Pruning the Network ("Joint" and "Special-Joint")

We first discuss the "Joint" proposal which consists of two stages. In the first stage, we find the two most similar candidate fractures as assessed by a similarity measure. We define a measure similar to the spatial distance measure described in Sect. 5.2. This measure incorporates the distances between the endpoints and orientations of fractures simultaneously and is termed the "product–similarity" measure,

$$d_{ps}(H^1, H^2) = \min(\theta_{13} \times \overline{p_1 p_3} + \theta_{23} \times \overline{p_2 p_3}, \theta_{14} \times \overline{p_1 p_4} + \theta_{24} \times \overline{p_2 p_4}) \qquad (5.9)$$

where $H^1 = [p_1, p_2]$ and $H^2 = [p_3, p_4]$ are two fractures from the fracture network and θ_{ij} and $\overline{p_i p_j}$ are the angles and absolute distances related to H_1 and H_2 as described in Figure **5.1**. The two most suitable candidate fractures correspond to the pair with the smallest value of the similarity measure. The selection process may fail when two fractures are co-aligned, whether or not they overlap. To

overcome this problem, we combine this measure with another similarity measure on fracture centres to obtain a general definition of similarity,

$$d_{pcs}(H^1, H^2) = (1 - \mu) \times d_{ps}(H^1, H^2) + \mu \times d(m^1, m^2) \tag{5.10}$$

where μ is a small positive number to handle the co-alignment problem (e.g.$\mu = 0.1$) and $d(m^1, m^2)$ is the absolute distance of the centres $m^1 \in H^1$ and $m^2 \in H^2$. In the second stage, the aim is to propose a new fracture and its parameters. The DD-transform is used to find the best choice among all possibilities. The rest of the procedure is the same as that of the split proposal, i.e. accepting or rejecting the proposal based on the acceptance probability. The "Joint" proposal is:

1. Apply the product-similarity measure, d_{pcs}, for pairs of fractures.
2. Choose a pair of fractures with the smallest value of the similarity measure.
3. Find the orientation and the coordinates of the midpoint of the new fracture by applying the DD-transform to the associated points of the candidate fractures.
4. Generate a value for the length of the proposed fracture from its distribution.
5. Accept or reject the proposal on the basis of Eq. (5.5).

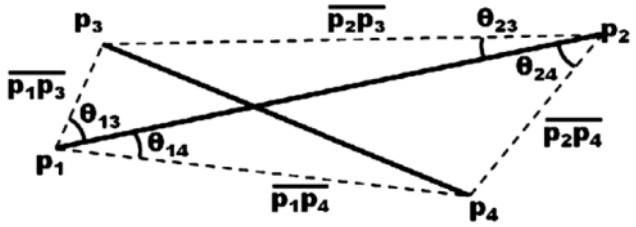


Figure 5.1: The lines $[p_1, p_2]$ and $[p_3, p_4]$ used in the product-similarity measure.

In joining two fractures, it may be necessary to consider criteria other than the similarity of lines. One of these is to combine fractures with small values of the associated point density, which is defined as the number of associated points of the fracture per unit length of the fracture; we refer to this as the "Special-Joint" proposal. The aim is to remove redundant fractures from the fracture network. This case is similar to the joint proposal in that the number of fractures is

decreased by one for each successful proposed move. First, the fracture with the smallest associated point density is selected. The selected fracture is removed from the network. The objective function value is calculated for the proposed combination based on A_P in Eq. (5.5). The proposal is accepted if A_P is less than a random number generated from the uniform distribution.

5.9 Experiments

A simulated dataset was constructed to test the proposed methods. A number of random points, here 70, were generated in the region of $[0,1] \times [0,0.5]$ with each point representing the midpoint of a fracture. For each point, values for length and orientation were drawn from their respective distributions. In this example, the distribution of length is lognormal, with mean 0.20 and standard deviation 0.13 and the distribution of orientations is Gaussian, $N(0,1)$. The values obtained for orientation are then truncated to the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$. The region displayed in Figure **5.2**a, is greater than $[0,1] \times [0,0.5]$ as the fractures are not truncated.

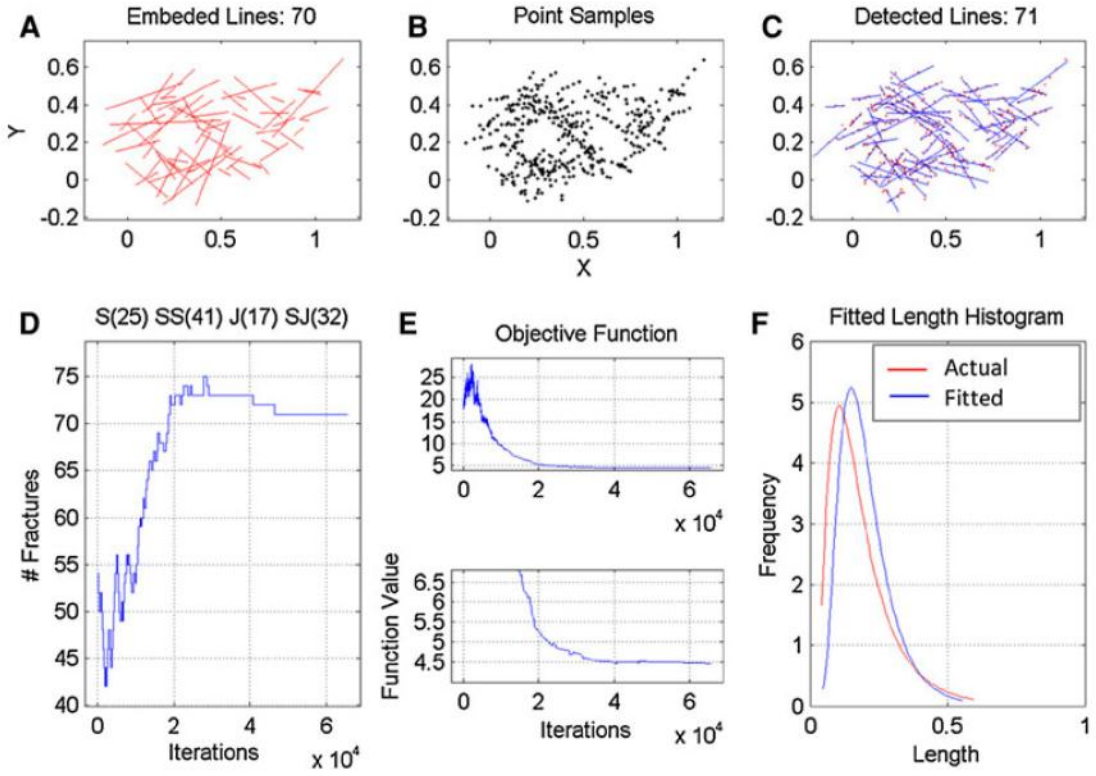


Figure 5.2: Results of the simulated dataset with 70 embedded lines; (a) simulated fractures; (b) point sampling with noise 0.01; (c) the final fitted fractures; (d) the number of

fractures versus the iteration number; (e) the objective function values versus the iteration number; (f) the models of the length histogram for the actual and fitted fractures

Figure **5.2**a is the "true" fracture network that is to be modelled. Figure **5.2**b is the point cloud simulating the seismic points associated with the fracture network, and Figure **5.2**c is the fracture model fitted using the proposed method. The points in Figure **5.2**b are obtained by using randomly sampled points of the fractures in Figure **5.2**a. The minimum and maximum number of points for the smallest and largest fractures are between 3 and 10 depending on fracture lengths; the longer the fracture, the greater is the number of points. Noise from a uniform distribution is added to the sampling points to create a dataset resembling reality. Figure **5.3** is the initial fracture model before optimization. For the initial model, some points, here 54, are selected from the point set at random as initial fracture centres; for each point, values of orientation and length are generated at random from their respective distributions. The final total number of detected fractures is 71 compared with 70 "true" fractures. The joint and split proposals are demonstrated in Figure **5.2**d where "S", "SS", "J" and "SJ" stand for "Split", "Special-Split", "Joint" and "Special-Joint". As shown in Figure **5.2**d, total numbers of 25 "Split", 41 "Special-Split", 17 "Joint" and 32 "Special-Joint" proposals were accepted. As expected from the global convergence property of SA (Kirkpatrick et al. 1983), after a certain number of iterations the number of fractures stabilizes; in this example, stabilization is achieved at around 30,000 iterations. The value of the objective function decreases and converges to a steady state as shown in Figure **5.2**e. In Figure **5.2**f, the length histogram of fitted fractures is compared with the initial distribution of the fracture lines in Figure **5.2**a; there is a close match between the two histograms. Figure **5.4**a shows the rose diagram of the orientations for the original "true" fractures and Figure **5.4**b shows the rose diagram of the fractures in the fitted fracture model. Although the orientation was not directly involved in conditioning the fracture simulation, the two still show significant similarity, given the noise on sample points and the few sample points on each fracture. To check that the model can generate a distribution of lengths similar to the length histogram model in Figure **5.2**f, we conducted 30 simulations with different sets of parameters for the lognormal distribution. In each simulation, noise from a Gaussian distribution was added to the actual parameters of the original dataset to

create a noisy set of parameters. The resulting length histograms are shown in Figure **5.5**c where the bold line is the actual histogram and the others correspond to the different simulations. Figure **5.5**a shows the number of fractures obtained after optimization versus the number of initial fractures, and Figure **5.5**b shows the function values in these 30 simulations. In Figure **5.5**a, two simulations co-locate with two other simulations (red circles).

5.10 Extension to 3D Applications

Several issues must be addressed to extend this work to 3D applications. First, fractures in 3D are no longer line segments and common representations of fractures in 3D include circular discs, elliptical discs, planar polygons or planes with infinite extent. The elliptical disc representation is the most common and is used in the research reported here, i.e. $H = (x, y, z, \alpha, \beta, \gamma, a, b)$, where x, y, z are the coordinates of the fracture centre point, α, β the dip direction and dip angle of the plane, γ the rotation angle of the major axis against the dip direction of the ellipse, and a, b the major and minor axes of the ellipse. Because of the tortuosity of fractures, this configuration, even with the 'best' fitted model, cannot intersect all seismic points, but the distance of the points to the fracture planes can be used as one of the criteria to assess the goodness of fit of the fracture model.

Second, the DD-transform must be extended so that it represents at least the centres and the orientations of the ellipses. For each candidate point in step 2 of algorithm 1, two distinct points in step 3 are required to propose a plane, hence determining the corresponding centre and orientation. Note that the rotation angle is not obtained from the plane information constructed by the three points in steps 2 and 3. The centre coordinates, the dip direction and the dip angle are available from the DD-transform in which a 3D array is constructed with the first dimension referring to the index of points, the second to the dip direction and the third to the dip angle. The other three parameters (the rotation angle, major axis and minor axis) are generated randomly from their respective distributions.

Third, the objective function in 2D applications is the sum of the spatial distances from fractures. The purpose of the objective function is not only to

minimize the distances of points to the fracture model, but also to condition the model to the fracture sizes. The extension of the objective function to 3D is not straightforward and requires an alternative formulation using two objective functions: one for minimizing distances of the points to the fitted fracture model and the other for minimizing the fracture sizes. The former function is simply a projection of the distances of the points to the fracture planes. The latter is defined as:

$$f_{AF} = \sum_{i=1}^{n} \frac{1 + a_i b_i}{1 + m_i} \tag{5.11}$$

where n is the number of fractures, a_i and b_i are the major and minor axes of the ith fracture and m_i is the number of points associated with the ith fracture ($m = \sum_{i=1}^{n} m_i$). This objective helps to achieve a network with the smallest possible number of fractures and a minimum amount (area) of fracturing (corresponding to the existing point cloud). The function is divided by $1 + m_i, i = 1, \ldots, n$, to achieve fractures with more associated points; the larger the value of m_i, the smaller is the value of f_{AF}.

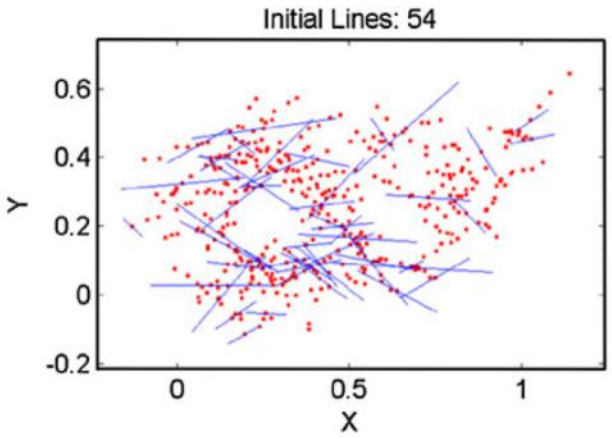


Figure 5.3: Initial map before optimization

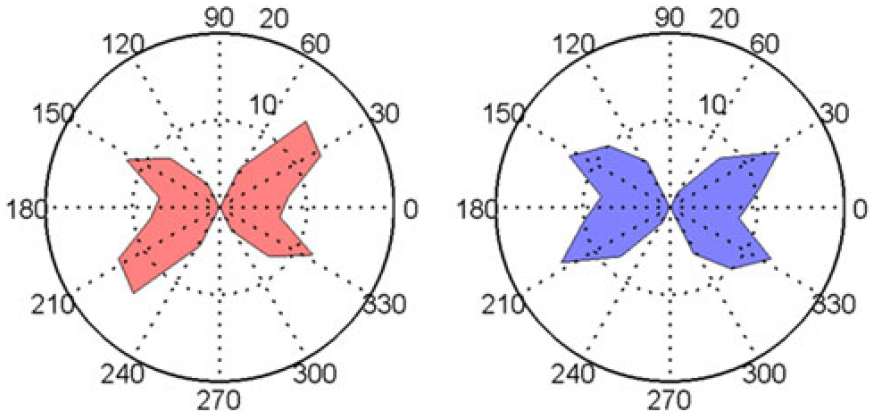


Figure 5.4: Rose diagrams for the simulated dataset: (a) actual lines; (b) fitted lines

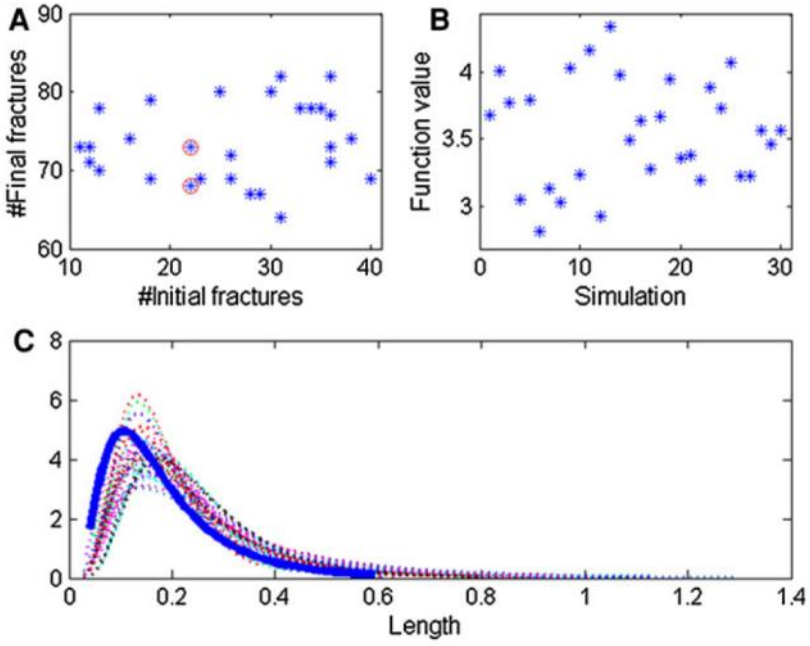


Figure 5.5: A summary of 30 simulations: (a) number of final fractures versus number of initial fractures; (b) function values after optimization; (c) actual (bold line) and fitted length histograms

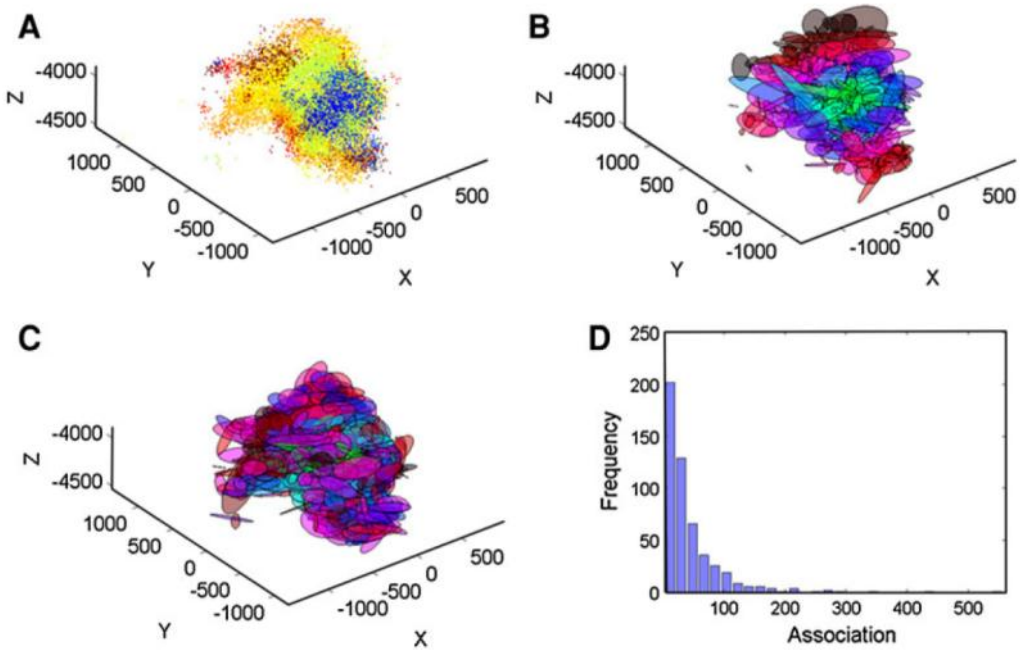


Figure 5.6: Results for Habanero dataset; (a) seismic point cloud; colours represent time domain of the seismic events; (b) initial fractures propagated from the borehole; (c) final fitted fractures; (d) distribution of point associations (i.e. number of points per fracture)

Fourth, the proposals are extended to 3D applications as follows. The centre and the orientation of fractures are perturbed by adding noise, using a random sample from a Gaussian distribution, while the fracture sizes (major and minor axes) are exchanged between two randomly selected fractures. The four proposals used for growing and pruning the network size can be easily adapted for 3D cases. However, some changes are required in the selection criteria for choosing a candidate fracture(s) and for determining parameters of proposed new fractures. In the "Joint" proposal, the 2D similarity measure is no longer applicable and the two closest fractures are chosen on the basis of the Euclidean distance of their parameters. In the "Split" and "Special-Split" proposals, the 2D selection criteria are equally applicable in 3D.

5.10.1 A Real Case Study: Habanero Reservoir Dataset

The Habanero wells are part of Geodynamics' HDR geothermal project in the Cooper Basin, South Australia. These wells have been drilled to depths of about 4,400 m below the surface or about 700 m into the bedrock where the temperature

reaches 250°C (Baisch et al. 2006). The dataset used in this study contains 23,232 seismic events covering an approximate area of 2.5 km^2. The absolute hypocentre locations of these events are shown in Figure **5.6**a.

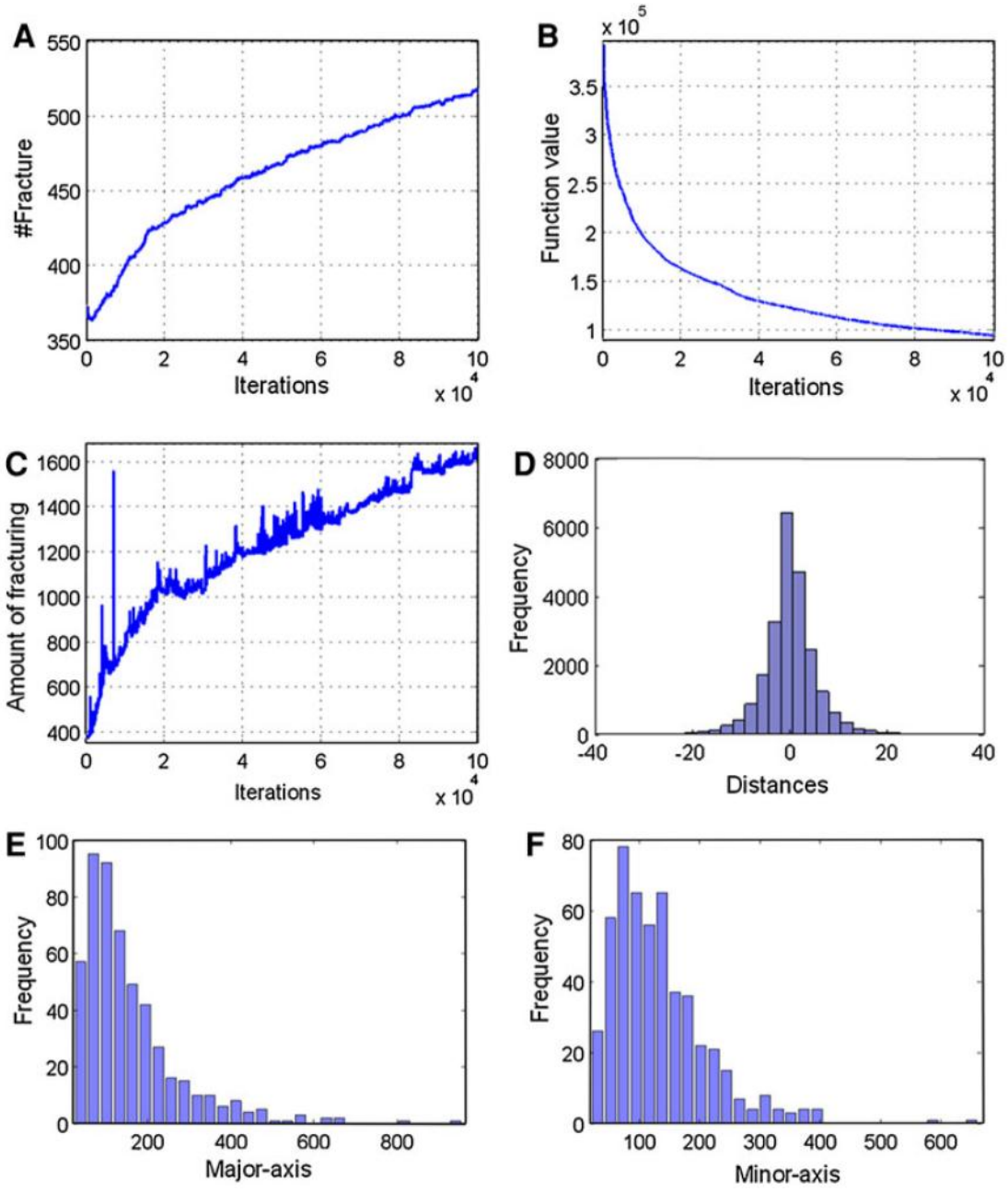


Figure 5.7: (a) Variation of number of fractures; (b) the total objective function value; (c) the amount (area) of fracturing during the optimization process; (d) distribution of the associated distances; (e) distribution of major axis; (f) distribution of minor axis.

The resulting fracture model after optimization (100,000 iterations of SA) is shown in Figure **5.6**c, which represents an optimal realization of the point cloud,

and Figure **5.6**b shows the initial fracture model. To create an initial map, the nearest points from a borehole are selected first as possible candidates for centres of new fractures. If the selected point is not in a pre-specified bandwidth of existing fractures, a new fracture is added to the network. This process continues until all points are visited and each has been assigned to a bandwidth of at least one fracture. The number of initial fractures here is 371. The number of fractures after optimization is 515 compared with 613 fractures in the work of Baisch et al. (2006) and Xu et al. (2013b), while retaining reasonable accuracy in terms of the associated distances and the point–fracture associations. The method is almost ten times more efficient than the MCMC model of Xu et al. (2013b). Figure **5.6**d shows the distribution of the point associations. From Figure **5.6**d, the two largest detected associations have more than 400 associated points, while in the work of Xu et al. (2013b) the largest fracture had an association of 393 points. Figure **5.7**a shows the variation in the number of fractures during the optimization process. This figure also shows the variation in the proposals ("Split", "Special-Split", "Joint" and "Special-Joint"). Figure **5.7**b shows the objective function value and Figure **5.7**c shows the amount (area) of fracturing. Figure **5.7**d shows the distribution of the associated distances, i.e. the distances of points from the fracture model. Figure **5.7**e, f shows distributions of the major and minor axes of the detected fractures, which have a lognormal distribution.

5.11 Conclusions and Future Work

We have developed a stochastic optimization method for fitting a fracture network conditioned by a seismic point cloud. The results from the simulated dataset were satisfactory in terms of the objective function value and statistics including the histogram of fracture length and the rose diagram of fracture orientations (for 2D).

The proposed method is general in the sense that it can be easily extended to 3D applications. The use of the DD-transform increases the accuracy as well as the efficiency of the modelling by introducing a locally optimal solution for fracture parameters. The results from the Habanero reservoir field data were satisfactory in terms of the number of iterations, the number of fractures and point association (for 3D).

Acknowledgments The work reported in this paper was funded by Australian Research Council (ARC Discovery Project) Research Grant Number: DP110104766.

6
Characterisation of Fracture Networks

Fracture networks can be characterised by their geometry, spatial arrangement and topological relationships. For a stable fracture network, fluid flow is a function of these defining characteristics. In a broader sense, if the intention (or even the outcome) of fluid injection (as in stimulation process) is to propagate fractures then it is more a response of the rock and the confining conditions. For simulation of fracture networks, the characteristic of a fracture network model can be defined somehow explicitly, however, for a given fracture network realisation a set of measures are required to determine and or to evaluate the network specifications. In this chapter, some known and some novel measures are introduced and discussed with the intention of providing an insight into the subject and purpose of characterisation of fracture networks.

6.1 Fracture centroid density

It is common practice to represent fracture locations in two- and three-dimensions by their centroids. The point densities of the resulting point patterns can be used, for example, to produce density maps (Fadakar-A et al. 2011). A common and recommended (due to its smoothing effect, efficiency and robustness) method for evaluating point densities is kernel density estimation (KDE, Wand and Jones 1995). KDE is covered extensively in the literature and there is a particular focus on the optimal determination of KDE parameters such as the choice of bandwidth, which has a significant effect on the resulting map (details in Duong 2004). As a proposal applying KDE to a set of fracture centroid points yields a new measure called the fracture centroid density (FCD, or DFC: density of fracture centroids). Figure **6.1** shows the FCD maps for a synthetic two-dimensional example of a fracture network.

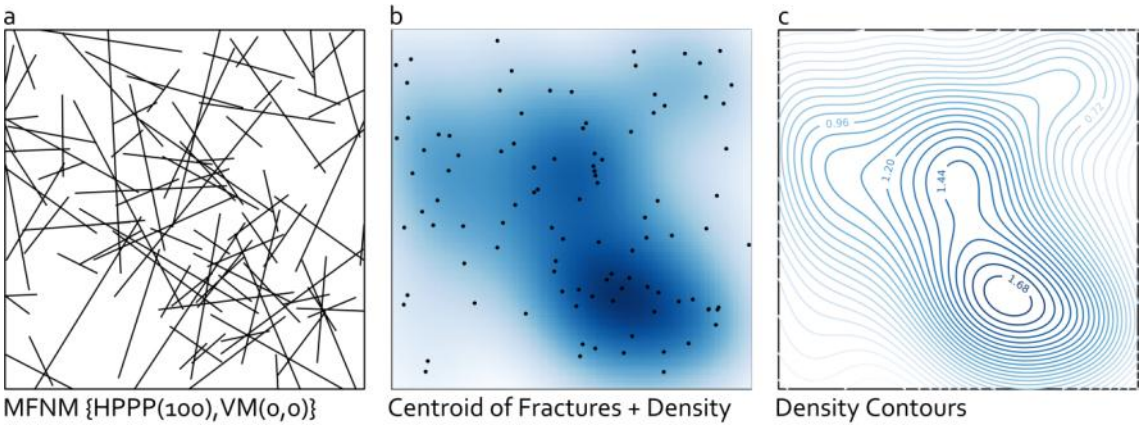


Figure 6.1: Fracture network density maps based on fracture centroid points (FCD map). A density map of this type can be seen as a quick and useful evaluation of concentration of fractures in the study region. It is, however, a biased estimation due to the simplistic representation of fracture lines as points.

Density maps, such as those provided by FCD, do not take into account the inter-connection among fractures in the network; these density methods are neither representative of intersection patterns nor connectivity characteristics of the fracture network. This remains true irrespective of the level of correlation between fracture density and fracture connectivity in any particular case. The example shown in Figure **6.2** demonstrates that a completely unconnected set of fractures, which would generate a void connectivity map, can nevertheless generate a regular density map. The fracture network in the example was generated using an inhomogeneous point density for fracture centres. Fractures were then generated using a truncated power-law distribution for lengths and a von-Mises distribution for orientations. Fractures were accepted only if they had no intersections with previously existing fractures in the set.

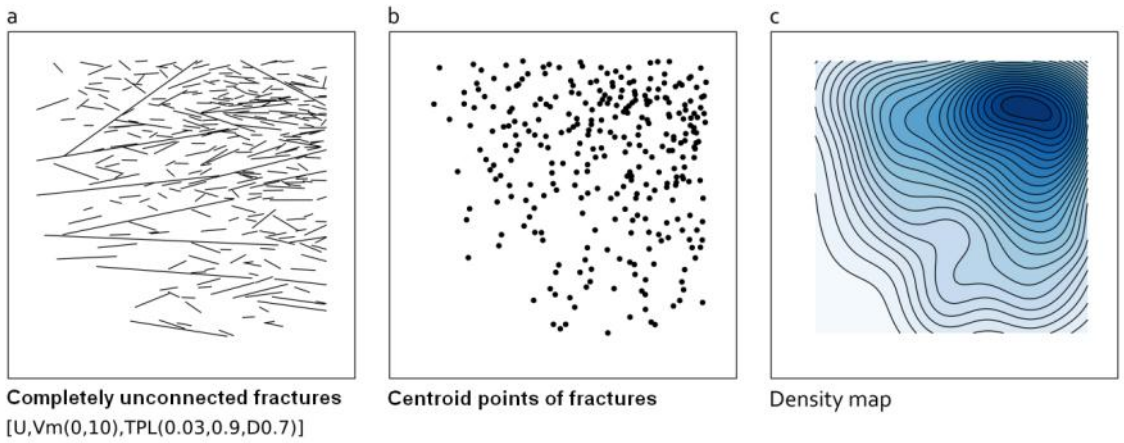


Figure 6.2: High fracture centroid density does not necessarily imply fracture intersections. In (a) despite its appearance (due to size of image resolution) there are no percolating sides. This example shows the serious shortcoming of using FCD for describing the connectivity and/or percolation state of the network.

A FCD map represents fractures solely by their centroid points and there is thus no information about length and orientation in the resulting density map. FCD is a simplistic form of the density of fractures that is only appropriate for a quick initial evaluation. The limitations of FCD can be addressed by using alternative measures such as the one proposed in Sect. 6.4.

6.2 X_f

The average number of intersections per fracture, X_f, is a traditional measure of the overall connectivity of a fracture network. To compute this measure, for each fracture in the network the number of intersections with other fractures is counted and the average of all of these values is the X_f measure, which is a single value for the entire network. An investigation was made to count the number of intersections per randomly generated lines instead of the fractures themselves. The resulting value was similar to that of the less-known P11 (see next, Sect. 6.3). Briefly, although the two measures X_f and P11 are similar, they differ fundamentally in concept and application. Note that X_f evaluates the connectivity of fracture networks (i.e., intersection between fractures must exist otherwise X_f is zero) while P11 evaluates the intensity of fractures (same as scanline sampling). That is, for a completely unconnected fracture network, such that in Figure **6.2**, the X_f value is zero while the P11 value is greater than zero.

6.3 Intensity group

Dershowitz (1992) proposed a list of six intensity measures known as the P (persistence) group including P11, P21, P31, P22, P32 and P33 of which P21 and P32 are the most widely used. The measures are summarised in Figure **6.3**.

Fracture Intensity				
		Dimension of measurement region		
		1 Line	**2** Area	**3** Volume
Dimension of fracture measure	**1** Number of fractures	**P11** Number of fractures per unit length of scan line / borehole (inverse spacing) [L^{-1}]	**P21** Number of fractures per unit area of trace plane [L^{-2}]	**P31** Number of fractures per unit volume of rock [L^{-3}]
	2 Dimension one less than that of measurement region		**P22** Length of fracture traces per unit area of trace plane [L^{-1}]	**P32** Area of fractures per unit volume of rock [L^{-1}]
	3 Dimension equal to that of measurement region			**P33** Volume of fractures per unit volume of rock [-]

Figure 6.3: Summary of fracture intensity measures proposed by Dershowitz (1992).

An example evaluation of P11 is shown in Figure **6.4** in which two scenarios for sampling lines have been implemented. From the Figure it can be seen the regular (i.e., single direction) sampling lines generate a lower P11 value (15.551) than the random scanlines (16.314). In fact, using a single direction or a restricted range of directions for scan lines yields a biased evaluation of P11. The resulting error can be significant for regularly oriented fracture networks as in the following example. Consider a case in which fractures are oriented in a limited range of directions. If sampling lines are aligned with those directions there will be no or minimum intersection between fractures and the sampling lines. On the other hand, random scanline sampling is unbiased with respect to the orientation of fractures.

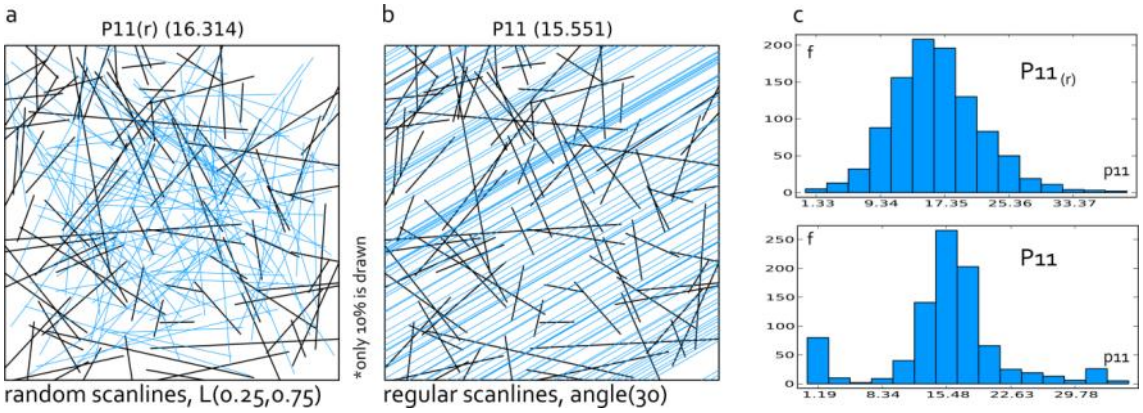


Figure 6.4: P11 using two systems of scanline sampling: random (a) and regular (b). Histograms of P11(r) and P11 are shown in (c).

P21 is demonstrated in Figure **6.5** for which two strategies were used. For the first strategy P21 was calculated as the total length of all fractures in the study region. For the second evaluation P21 was calculated as the average of the values yielded by Monte Carlo sampling (Robert and Casella 1999) of smaller squares (see Figure **6.5**b). For a sampling square of 0.1×0.1 within a 1×1 square study region, the resulting P21 value is 34.513 (approach 2) compared to 31.757 as of the entire region (approach 1).

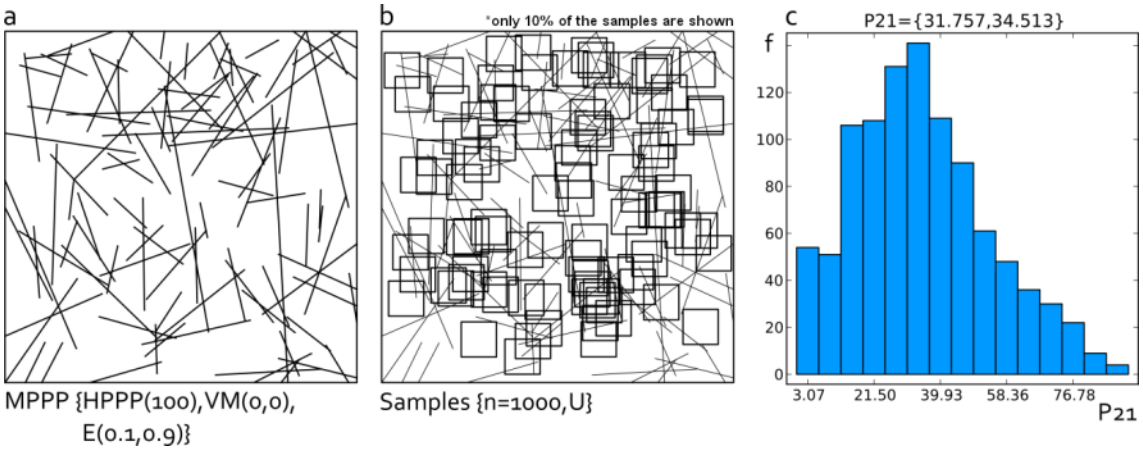


Figure 6.5: P21 using Monte Carlo simulation. A number of 1000 square samples of size 0.1×0.1 were taken. In (c) in the title of histogram of P21 values, 31.757 is for entire study region while 34.513 is average of 1000 samples.

The effect of variation in the sample size was investigated using a simulation with sample sizes varying from 0.1 to 0.5 within a study region of size 1×1 for ten realisations, each with 1000 samples. The results are shown in Figure **6.6**.

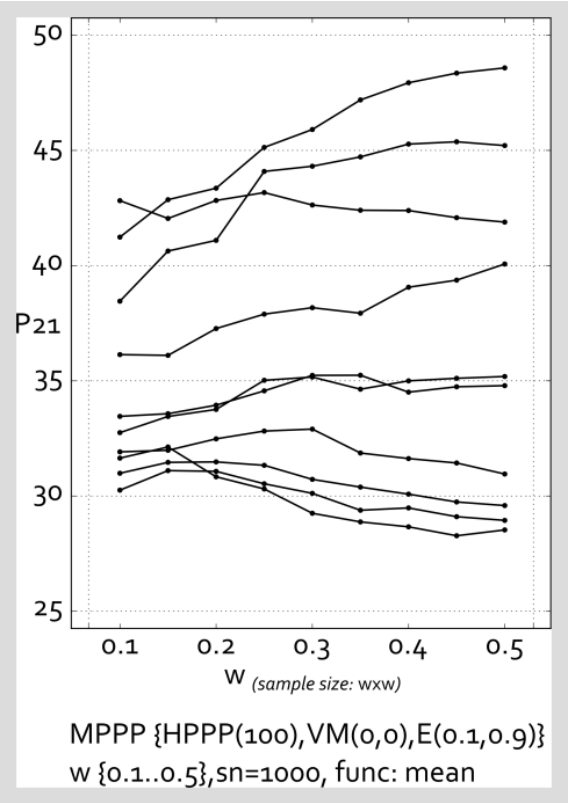


Figure 6.6: It appears that the size of sample (w) influences the calculated P21 value. However, the sign of the influence varies among the different realisations, suggesting that the proposed relationship is strongly associated with each realisation. In the figure, each data point of each curve is produced by averaging 1000 samples. Ten realisations were used.

From Figure **6.6** there appears to be a clear correlation between the sample size and the resulting P21 value; note, however, that the sign of the correlation varies among the realisations.

6.4 Fracture density

As a new proposal, the density of fractures in two-dimensional fracture networks can also be estimated by means of cell sampling on a grid as follows. The fracture density for any cell (e.g., square in Figure **6.7**) in the grid covering the entire study region is calculated as the sum of the number of fractures wholly contained within the cell and the number that intersect the boundary of the cell. This produces a matrix of fracture density (Fn), which can be mapped as the example shown in Figure **6.7**. Moreover, implementation of various cell sizes results in multiple Fn

maps on which *E-Type* measures can be used to calculate a Generalised Fn (GFn). These measures can be readily extended to three-dimensional fracture networks for which the sampling cell would be a cube.

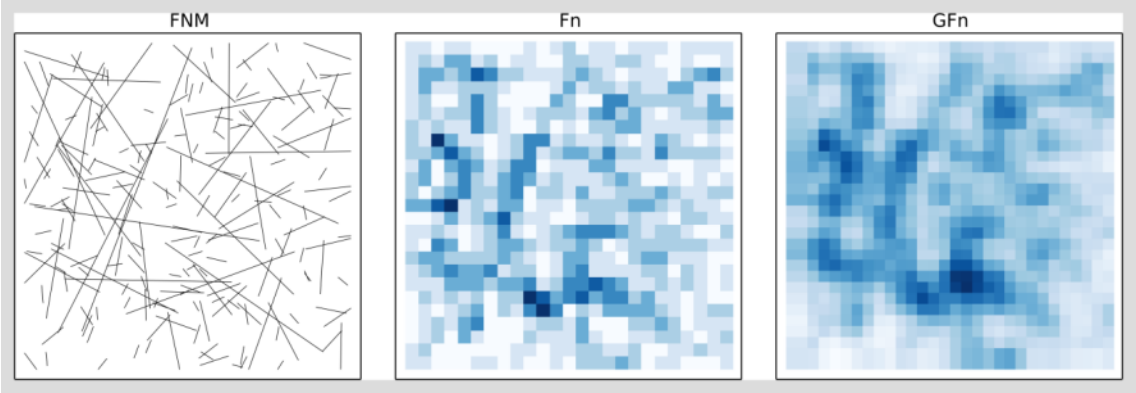


Figure 6.7: Fracture Density calculated for each cell in the grid by counting the number of fractures wholly contained within a cell and the number that intersect the boundary of the cell. GFn is the Generalised Fn computed for grid sizes in the range [3, 25], i.e., cell size varying from 1/9 to 1/625 of the area of study region.

6.5 Largest empty circle

The largest empty circle (LEC) in a region containing points is the largest circle that does not contain a point, i.e. it is a measure of the largest empty space in the region. A motivation for finding the LEC in a study region would be the placement of a nuclear reactor as far away as possible from surrounding cities. As a proposal in fracture network applications, LEC can roughly approximate the largest size of intact rock mass; suggesting applications for designing underground cavities, tunnels, buildings and so on. Figure 6.8 demonstrates the LEC for a set of points representing the centroid of fractures. Intuitively, a similar concept can be developed for three-dimensional fracture networks for which the objective would be the largest empty sphere. Note that, because of intrinsic assumptions, these measures are approximate. One such assumption is that fractures can be adequately represented by their centre points, which is not entirely realistic for rock masses. Another approximation is the use of a circle (sphere) to simplify the area (volume) of intact rock. The next two sections Sects. 6.6 and 6.7 (Distance map) provide ways of addressing these two shortcomings.

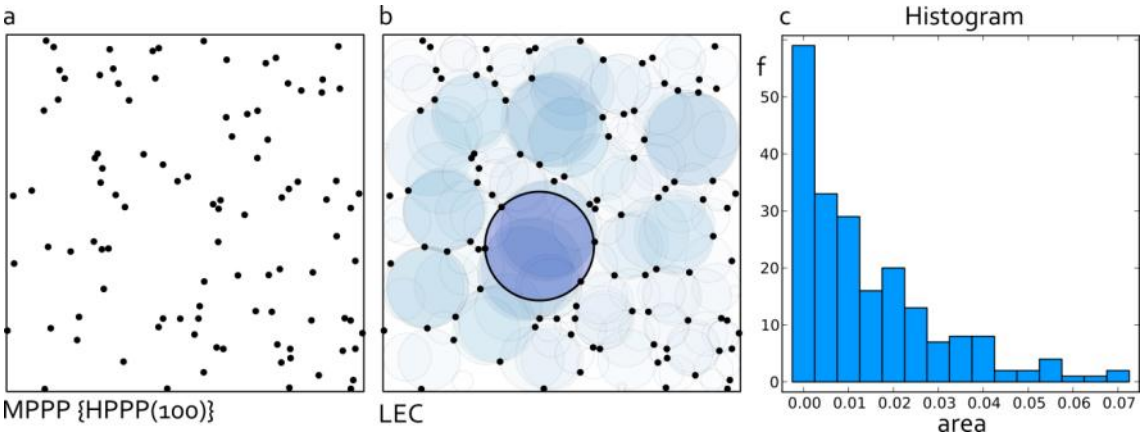


Figure 6.8: LEC analysis. The largest one is highlighted. This measure is useful to determine isolated areas in the study region according to a distance of interest. Note that the edges of the study region have also been considered as constraints. The exponential distribution of the areas is apparent as shown in the histogram on the right.

6.6 Largest empty convex-hull (inner-convex-hull)

Largest empty convex-hull (LECH or ICH) analysis is used to determine areas in the study region with the largest convex polygon that does not contain a point. Intuitively, this measure is more precise than LEC (see previous section) in providing the largest intact areas in rock masses. A precise solution is presented in the next section.

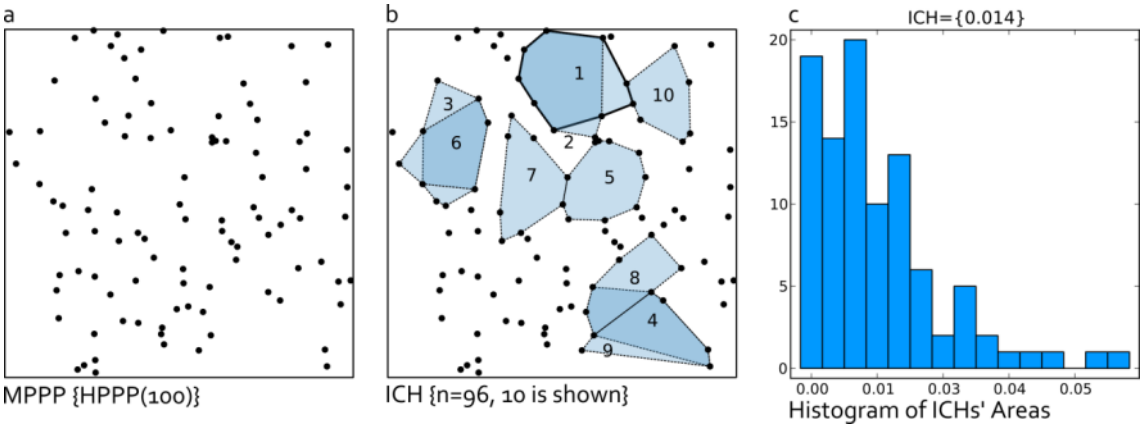


Figure 6.9: Largest empty convex-hulls. Similar to LEC in its application but provides a much more realistic measure of empty space. Note that for any fitted convex-hull a maximum area circle can easily be found.

6.7 Distance map

The distance map is a standard tool in image processing (Kimmel 2003). It is computed as a Euclidian distance field around a target object (here fracture trace lines). It is a simple function $Tp: \mathbb{R}_2 \rightarrow \mathbb{R}_+$ which assigns values to all points in the domain such that $T_p(q) = \inf_{p \in P} \mathbf{d}(p, q)$, where $\mathbf{d}(p, q) = \|p - q\|_{L_2}$ is the Euclidian distance between two points p and q. As a proposal for application of distance map in characterising fracture networks, as for example implementation stage in two-dimensional fracture networks, fracture traces are first discretised into pixels with a desired resolution (small enough to preserve the overall structure of the fracture network). For the example shown in Figure **6.10** the resolution was 0.001 compared to the extent of study region 1×1. The size was chosen so as to distinguish dense areas. The distance function results in a matrix of distances (Figure **6.10**b). A contour map of the distances is also shown in Figure **6.10**c. The distance field (map) analysis provides a simple and useful view of intact areas in the study region.

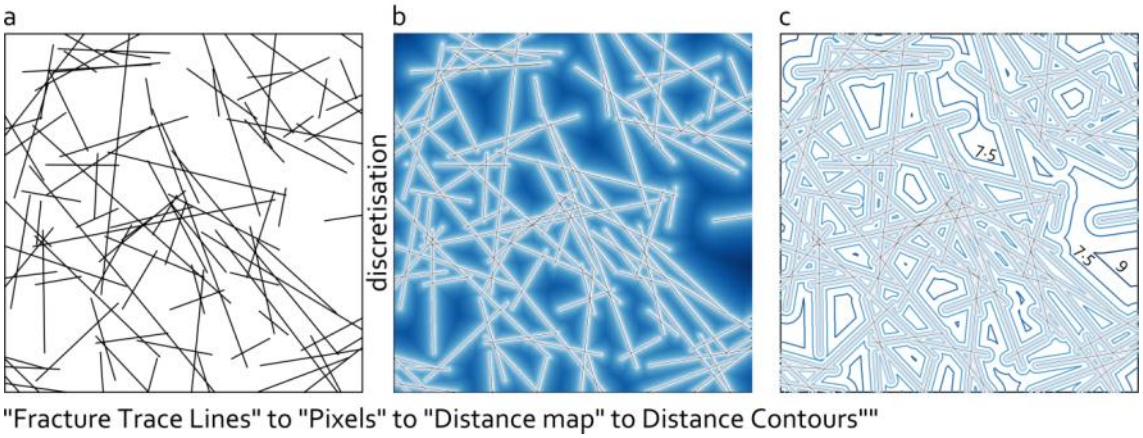


Figure 6.10: Distance map based on the distance from fracture trace lines. The darker the shade of blue the farther the location is from a fracture. This analysis may have applications in ranking a study region for safety issues. In (c) the contour values are the logarithms of the distance values.

6.8 Buffer effect

The buffer effect is widely used in Geographic Information Systems (GIS) to determine an affected space around a target object, such as restricted areas around

a road, railway and so on. It was originally a product of the dilation (a mathematical morphology function, see Sect. 2.1.11) of an object by a smaller, and often simple, geometrical shape such as a circle (2D) or a sphere (3D). It can also be produced using a distance map (Sect. 6.7). As shown in Figure **6.11**c buffers may overlap, which may indicate a potential new fractured zone due to the expansion of fractures (e.g., due to stimulation). The expansion is considered in both length and lateral extent. The buffer map can be used to evaluate the probability of expansion of the fracture network due to a stimulation process. That is, larger and multiple overlapping areas suggest a higher probability of newly expanded areas in the network. Finally, the size of the buffer can be associated with parameters of each individual fracture such as aperture and length.

6.9 Convex-hull

Fracture clusters are determined by means of intersection analysis. An isolated fracture is one that has no connection to other fractures. Isolated fractures are not involved in fluid flow modelling. A fracture cluster can be bounded by a convex-hull defining the minimum convex area surrounding it. Overlaps (i.e., shared areas), as shown in Figure **6.11**b, show regions of potential expansion of multiple clusters similar to that explained in the previous section on the buffer effect. The convex-hull and the buffer effect indicate very similar potential areas for expansion.

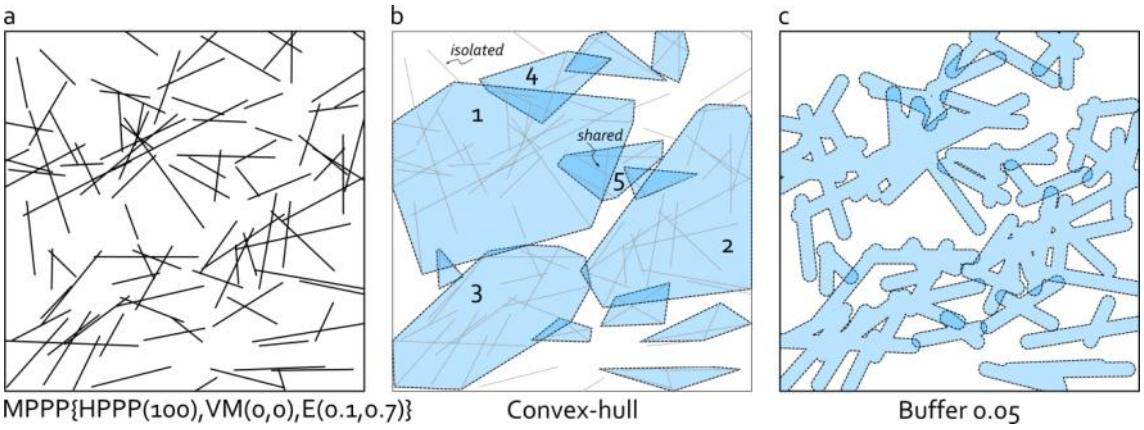


Figure 6.11: Convex-hull and buffer effect applied to a fracture network. Note that the buffer effect is applied only on clustered fractures assuming that the isolated fractures are not affected by expansion mechanisms.

6.10 Block area

Rock blocks are duals to fractures in fractured rocks. The analyses of rock block statistics, geometry and displacement are important especially in rock mechanics and rock engineering, where the safety of underground works are closely associated with the stability of rock blocks. Investigations show that the size of rock blocks (i.e., area for two-dimensional cases) has a distribution with a long positive tail. As shown in Figure **6.12** for GFNM fracture network models the resulting histogram of areas is exponential in form with no upper tail whereas for VFNM upper and lower tails are present although large areas are rare in either tail (see Figure **6.12**).

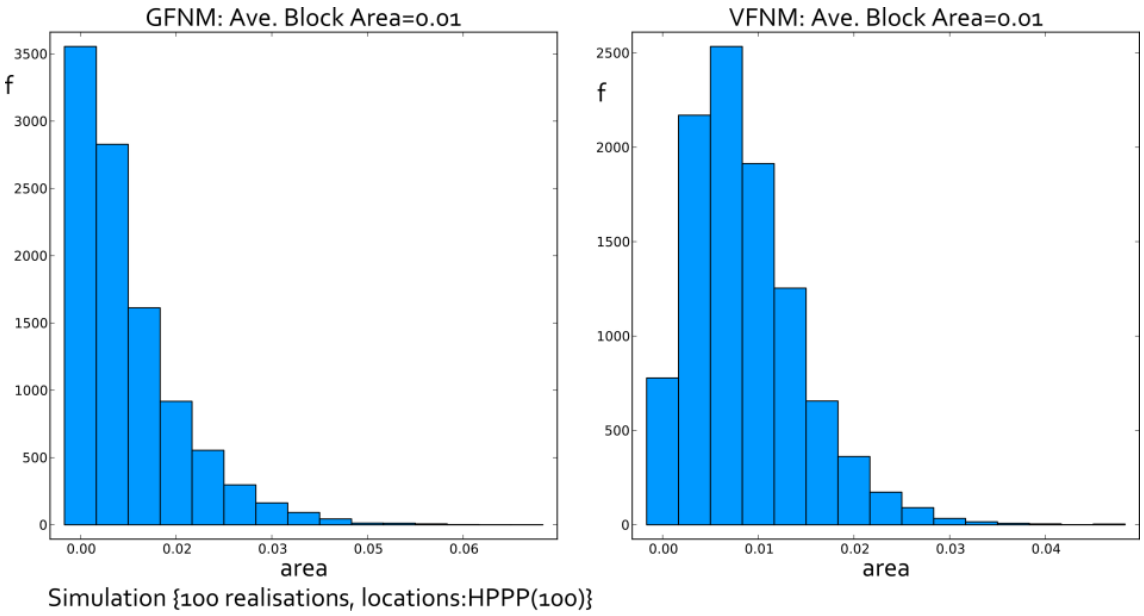


Figure 6.12: Histograms of block areas for GFNM and VFNM. Smaller area blocks are more dominant in GFNM compared to VFNM.

6.11 Backbone density

Any fracture network, if not fully isolated, has an internal structure called a *backbone* (also known as a *skeleton*, Priest 1993), which is the determining structure for fluid flow through fractures. A backbone structure of an FNM can be extracted by 1) removing isolated fractures, and 2) trimming free-end fractures to the nearest intersection point. Figure **6.13** shows the backbone of an FNM and the

density map of centres of segments of the backbone. Due to significance of backbone structure on flow the density of centroids of backbone elements (shortly BBD) is proposed as another useful measure. BBD can also be seen as an important tool for evaluating the domain for reaction between fractures and the fluid in a qualitative manner. Denser areas in BBD suggest more involvement in the flow regime for those areas. One may also remove any isolated clusters from the structure of backbone to decrease the computational effort in evaluating fluid flow. Isolated clusters are those that have no connection to boundary conditions, inlet or outlet.

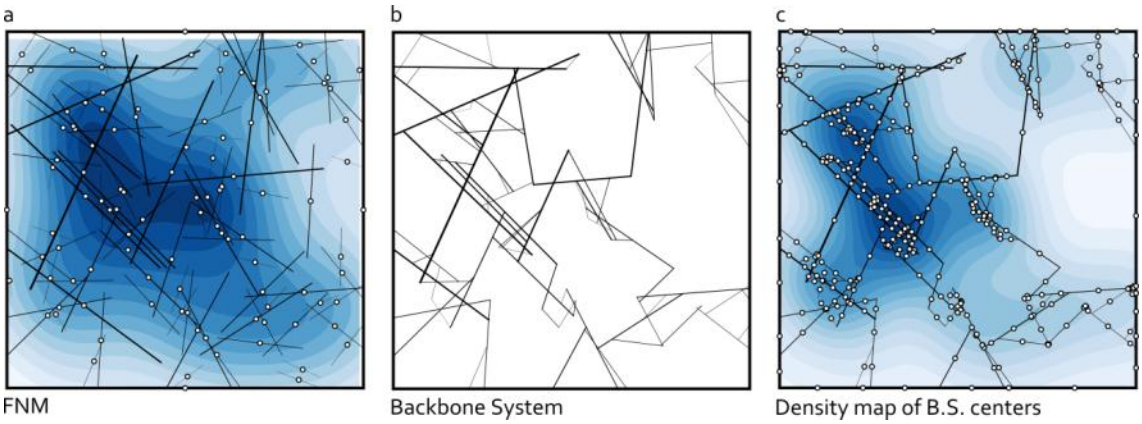


Figure 6.13: Backbone structure of an FNM. (a) FNM; (b) backbone; (c) density map of centres of backbone line segments.

6.12 Block centroid density

The blocks in a rock mass can also be represented by their centroid points, which makes them amenable to point density mapping techniques (Figure **6.14**). As a proposal an application of block centroid density (BCD) analysis would be for safety analysis in underground cavity design. Dense areas of blocks suggest higher potential for movement (instability) and so require additional support (e.g., bolting).

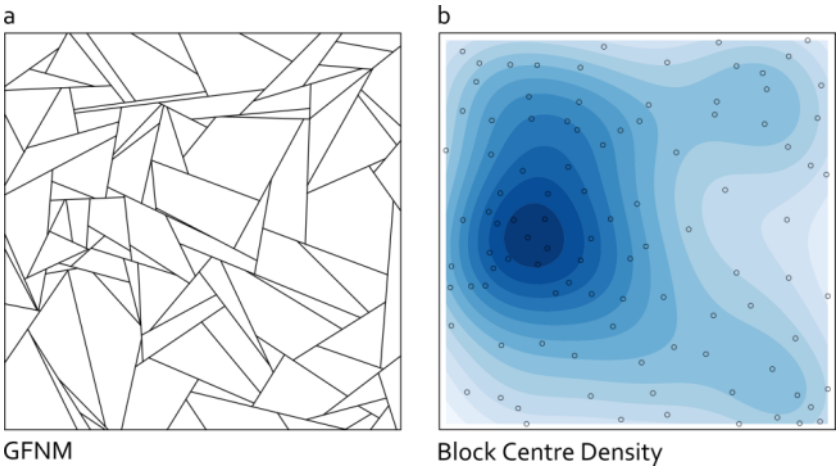


Figure 6.14: Density map of block centres in an FNM, which is useful in identifying blocks of smaller area blocks in rock masses.

6.13 Intersection analysis

Determination of intersections between lines in two-dimensional fracture networks and between polygons in three-dimensional cases is the first stage in the connectivity analysis of fracture networks. The various ways in which two lines can intersect are illustrated in Figure **6.15** and intersection algorithms must consider all of these possibilities. Computer-code implementations must be as robust as possible and error-free. Various techniques have been proposed for exact computation to avoid round-off errors due to floating point representation limits in computer systems. The analysis of intersection (inter-connection) of fractures in fracture networks is an important stage in characterising the network. For example, connectivity analysis follows fracture clustering as a direct output of intersection analysis. Connectivity is a key factor in creating pathways in geothermal energy systems for fluid flow through fracture networks.

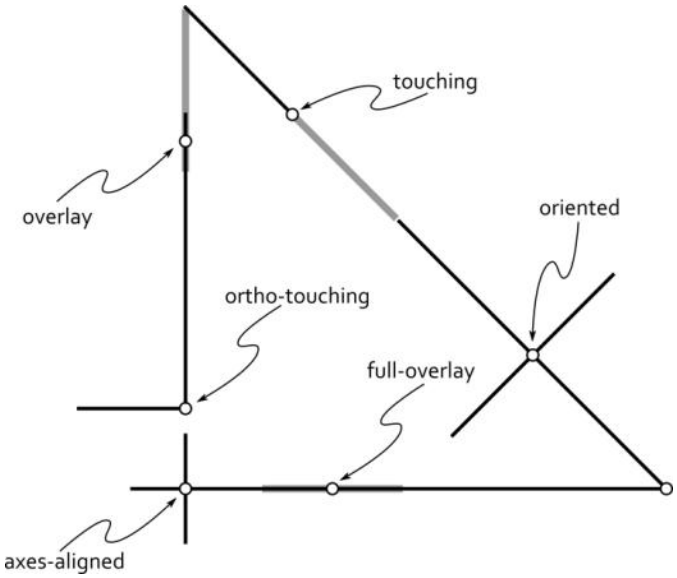


Figure 6.15: All possible ways in which two lines can intersect. In the overlaying cases it is better to choose the centre of the overlapping segment although other choices are also valid. For example, when using intersection analysis for segmenting the overlaying lines both endpoints are reported.

6.13.1 Fracture clusters

Fracture clusters i.e., groups of inter-connected fractures, are produced by the intersection between fractures. A cluster consists of at least two fractures. The existence of clusters in a fracture network may be associated with the heterogeneity of a fracture system (La-Pointe and Hudson 1985). Examples of fracture clusters determined on a synthesised fracture network are shown in Figure **6.16** in which the largest cluster has 24 members (fractures, elements) from the total number of 100 fractures. Note that despite such a significant proportion (24%), the area covered by the cluster is limited to a small portion of the study region. The area covered by a cluster can be quantified by using various tools such as the convex-hull or the buffer effect. It can therefore be concluded that the cardinality of a cluster is not highly correlated with its areal coverage.

The extraction of fracture clusters is a step in intersection analysis. The clustering information is then used to determine the connectivity of the fracture network. This is very useful for a rapid evaluation of the connectivity between a newly added fracture or a set of fractures (e.g., support edges) and the existing fracture network. Such a task is common in connectivity analysis including the

connectivity index (Chap. 7 and Xu et al. 2006) and the connectivity field (Chap. 7 and Fadakar-A et al. 2014) between supports and the fracture network.

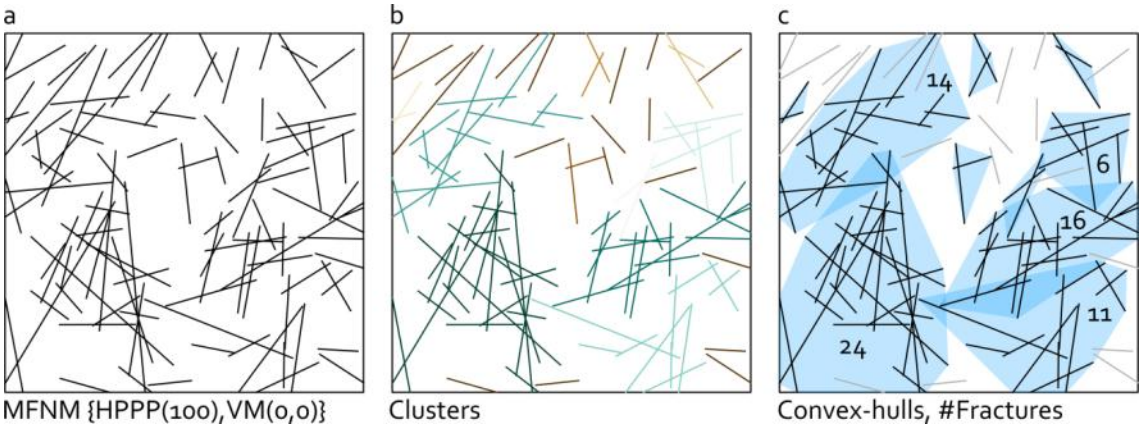


Figure 6.16: Application of intersection analysis results in groups of fractures called fracture clusters. The size of a cluster can be defined as its number of member fractures (cardinality) or the area that the convex-hull of a fracture cluster covers (coverage).

The application of fracture intersection analysis and clustering described can be extended to three-dimensional fracture networks with no additional concepts. An example of a realistic three-dimensional fracture network is shown in Figure **6.17**a for which the first three largest clusters with cardinalities of 94, 25 and 22, and the second three largest clusters with cardinalities of 13, 11 and 10 are shown in Figure **6.17**b and 6.17c), respectively.

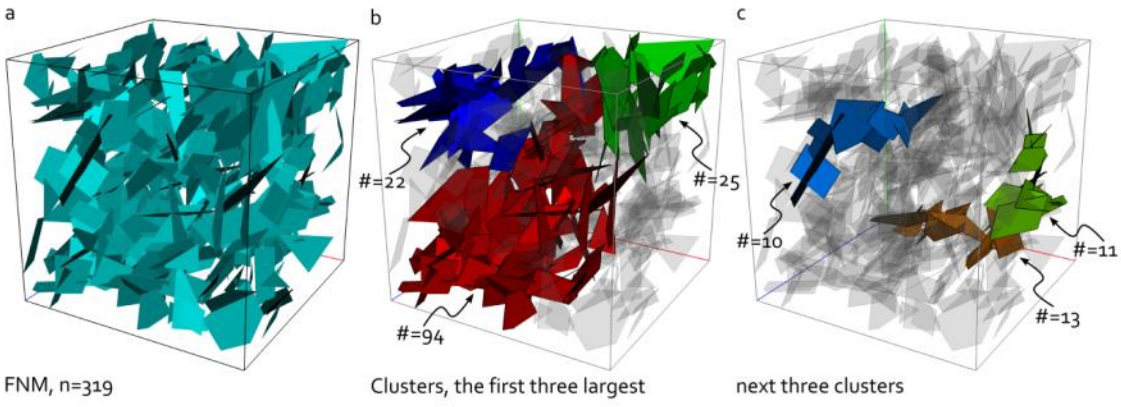


Figure 6.17: Three-dimensional fracture network and fracture clusters. In (b) the first largest clusters are shown in decreasing order in red, green and blue. In (c) the next three largest clusters are shown. The cardinalities of the value clusters are shown next to them.

Fracture clusters can be used for various purposes. For example, as shown in Figure **6.18**, the pipe model constructed on the first three largest fracture clusters,

Figure **6.17**b, provides an overall representation of fluid pathways in three-dimensional fracture networks. Combining fracture clusters with a pipe model is very useful in understanding possible pathways in the fracture network for fluid flow between any pair of locations. Where percolating sides exist such a representation (clustered pipes) helps to determine the domain in which fluid flows.

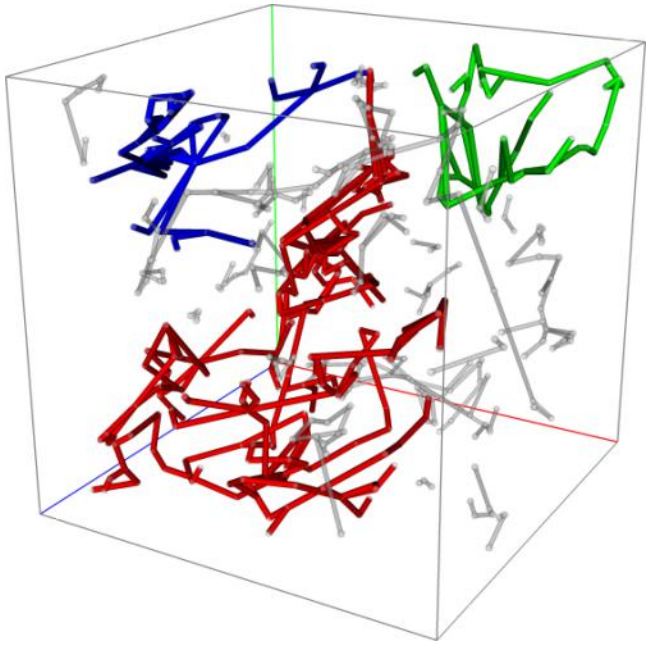


Figure 6.18: Pipe model constructed for the three-dimensional fracture network model shown in Figure **6.17**. The first three largest fracture clusters are highlighted. The pipe mode clearly exhibits the possible domain for fluid transport for each fracture cluster.

6.13.2 Intersection density

As described in Fadakar-A et al. (2011) the analysis of intersection points between fractures is an important measure that is closely related to the connectivity of fracture networks and so can be associated with the response of the fracture network to fluid flow.

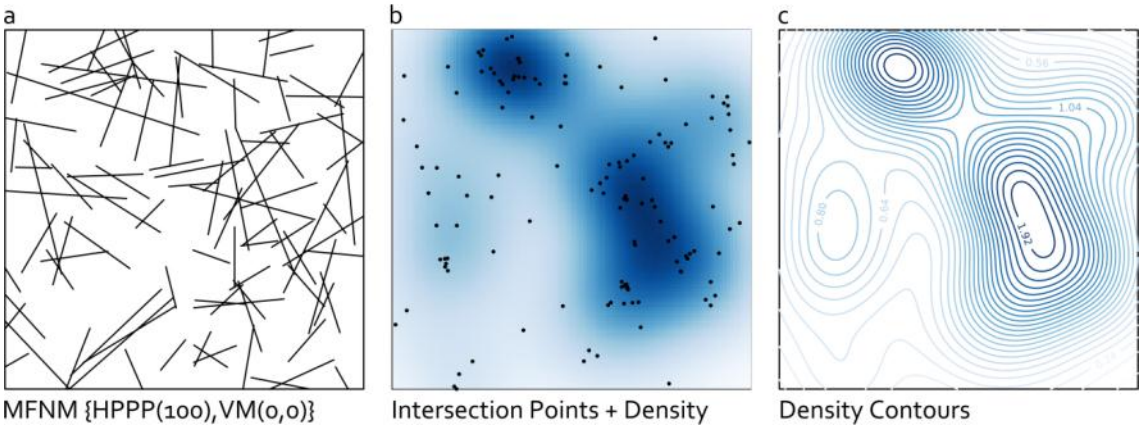


Figure 6.19: The intersection points between fractures are used for density mapping.

6.13.3 Extended intersection density

The infinite extension of the length of fractures generates a fully connected fracture network for which all intersection points can be determined and a density map can be generated (Figure **6.20**). This proposes a new measure called extended intersection density (EID).

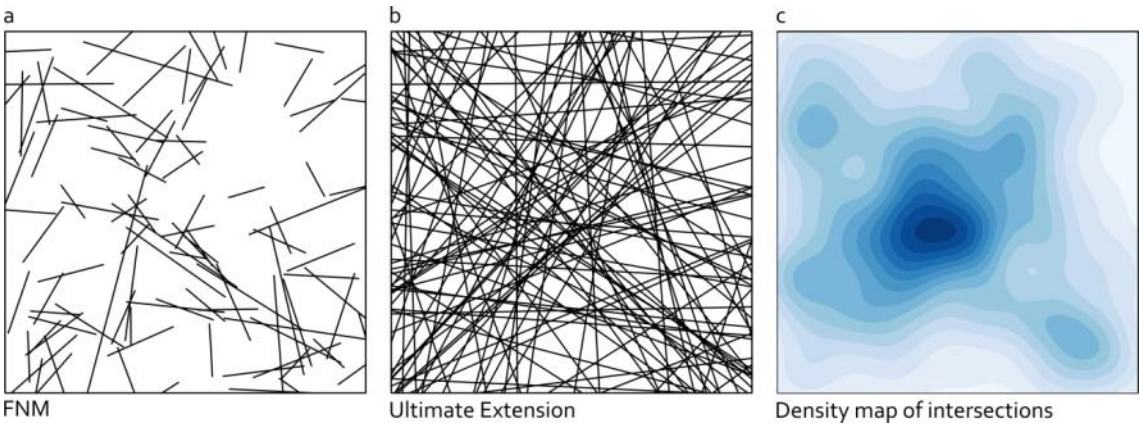


Figure 6.20: Intersection Density maps for an extended FNM. Every fracture trace has been extended to reach the boundary of the study area.

The resulting measure is a density map called the extended intersection density (EID) map.

6.13.4 Inter-connectivity

Inter-connectivity in a fracture network can be defined between at least two sets of fractures that are distinguished due to some criteria such as orientation. For

example, in the following example two fracture sets are defined on the basis of having an acute angle less than 45 degrees or higher. The inter-connectivity measure, as defined in Rouleau and Gale (1985-1987), is calculated as:

$$I_{ij} = \frac{l_i}{S_i} \sin y_{ij} \qquad (i \neq j) \tag{6.1}$$

for two fracture sets indexed by i and j. In general, for n fracture sets the measure is defined as:

$$I_i = \sum_{j=1}^{n} \frac{l_i}{S_i} \sin y_{ij} \qquad (i \neq j) \tag{6.2}$$

where l_i is the mean length of fractures in each set, S_i is the mean spacing between fractures in each set and y_{ij} is the mean angle between every two sets.

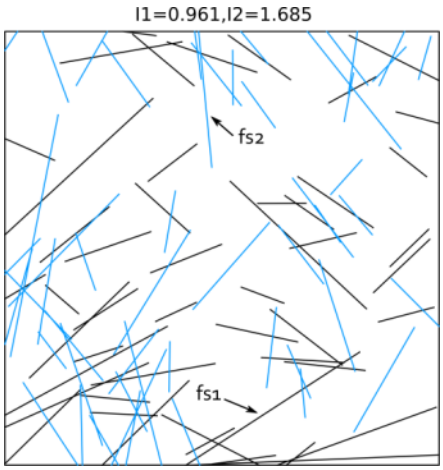


Figure 6.21: Inter-connectivity (I_i) between two sets of fractures in a fracture network. I1 in the title means $i = 1$ i.e., fracture set 1 (fs1) and similarly for fs2. As shown, this measure is not transitive; depending on the choice of i two different values are calculated as 0.961 and 1.685.

According to Rouleau and Gale (1985-1987) and Lee (1990) this measure is useful in indicating the importance of every fracture set individually in respect to its hydraulic role in the network. Figure **6.22** demonstrates the values of the measure for three fracture sets defined by orientation classification. It can be seen

that the measure has higher values for cases in which the first set is the class 2 (Figure **6.22**).

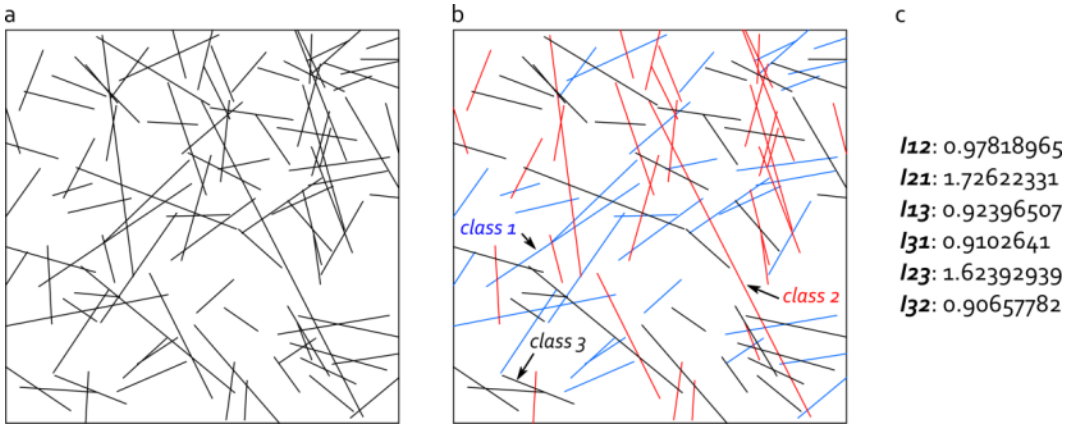


Figure 6.22: Inter-connectivity measured between three fracture sets (orientation classes).

6.13.5 Fracture normal intersection density

If the normal vectors to fracture lines (planes in three-dimensions) are generated in association with the length of fractures (Figure **6.23**), a new set of intersecting points between the normal vectors can be derived on which density evaluation can be applied. Further inspection of the fracture network and the resulting normal intersection density map may suggest areas with high mechanical activity. This suggestion may be justified to some extent by considering the normal vectors in the σ_3 direction (lowest principal stress direction) for every fracture in the network. This proposal is related to the concept of *Fabric Tensor* in rock engineering (see also Sect. 6.14 for a related discussion).

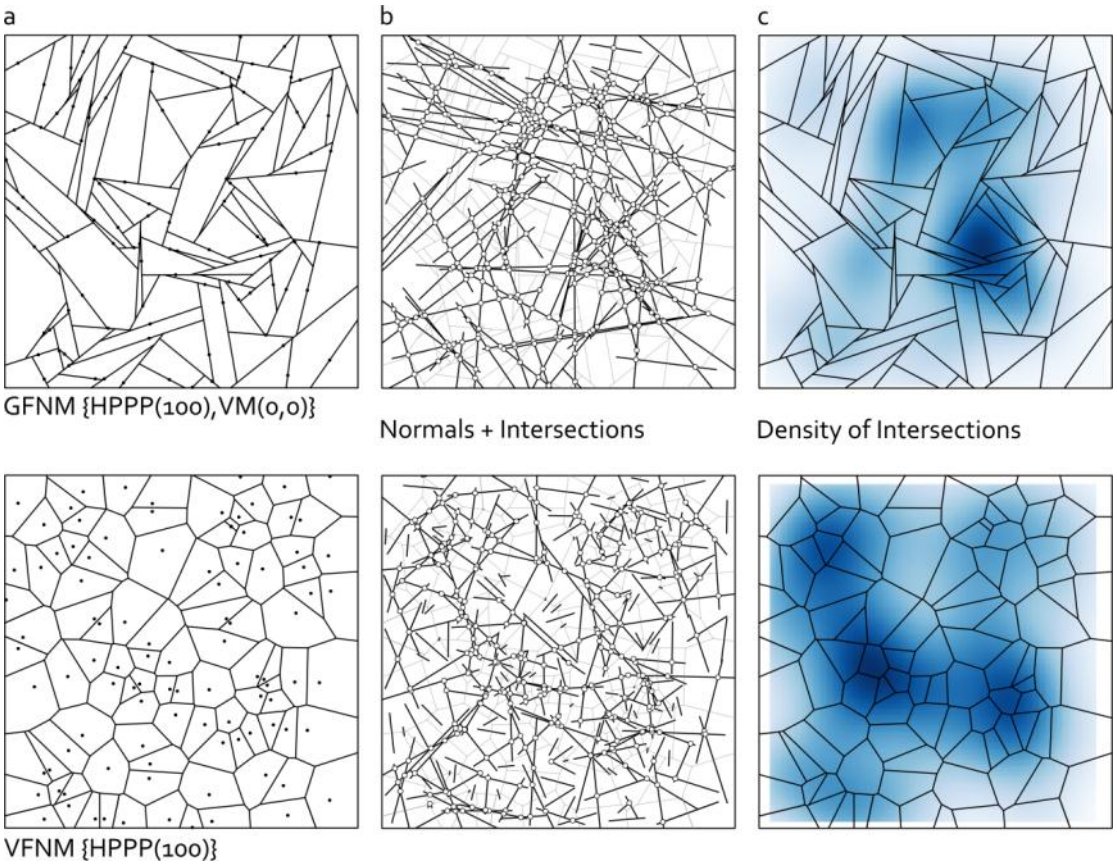


Figure 6.23: Density maps of intersections between normals of fractures for GFNM and VFNM. The normal lines are the same size as the fractures. The VFNM model suggests that NID is associated with the BCD whereas GFNM is not.

6.14 Effects of stress field

Fracturing due to brittle failure in rock can generate different fracture patterns depending on the relationship between the differential stress $(\sigma_1 - \sigma_3)$ and maximum strength (T) of the rock during fracturing (Cosgrove 1998). The patterns of extensional failures (i.e., $\sigma_1 - \sigma_3 < 4T$) are shown in Figure **6.24**, for example. The corresponding simulated fracture pattern for each stress state is conducted simply by adjusting the density of points (λ) and the Fisher dispersion factor (κ).

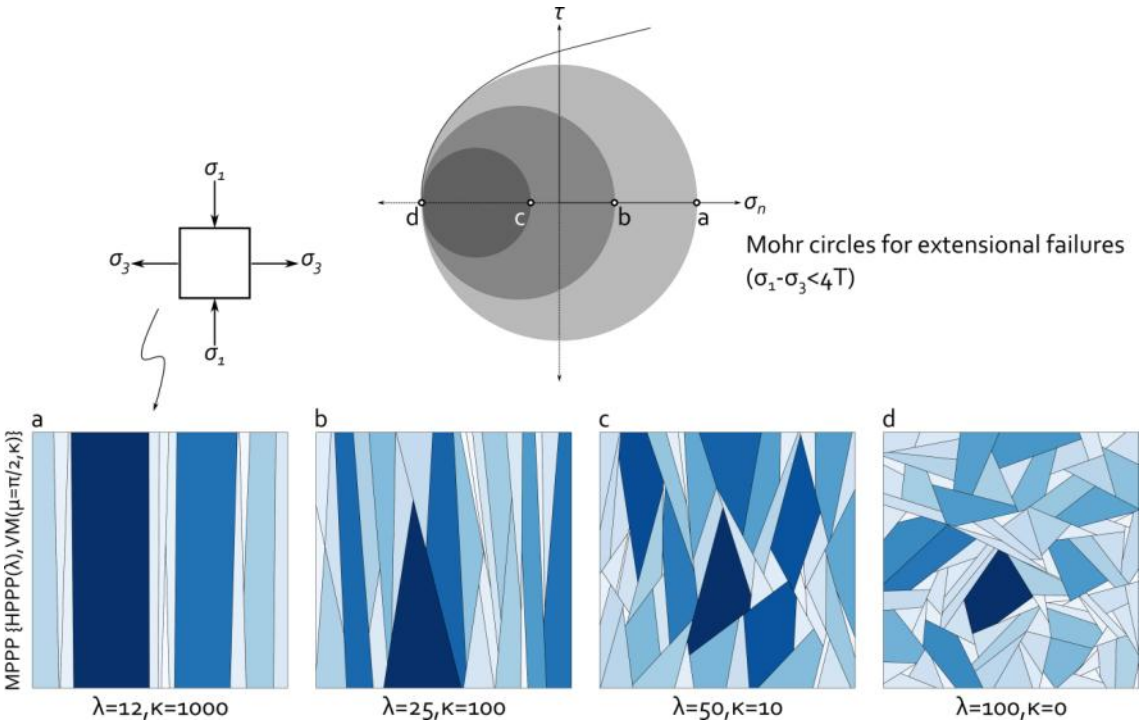


Figure 6.24: Mohr circles for extensional failures (e.g., due to hydraulic pressures) and associated fracture patterns. Labels "a" to "d" on Mohr circles correspond to patterns shown on sub-figures (a) to (d), respectively.

The effect of variation in the setting of principal stresses on the resulting fracture patterns in three dimensions is also shown in Figure **6.25**.

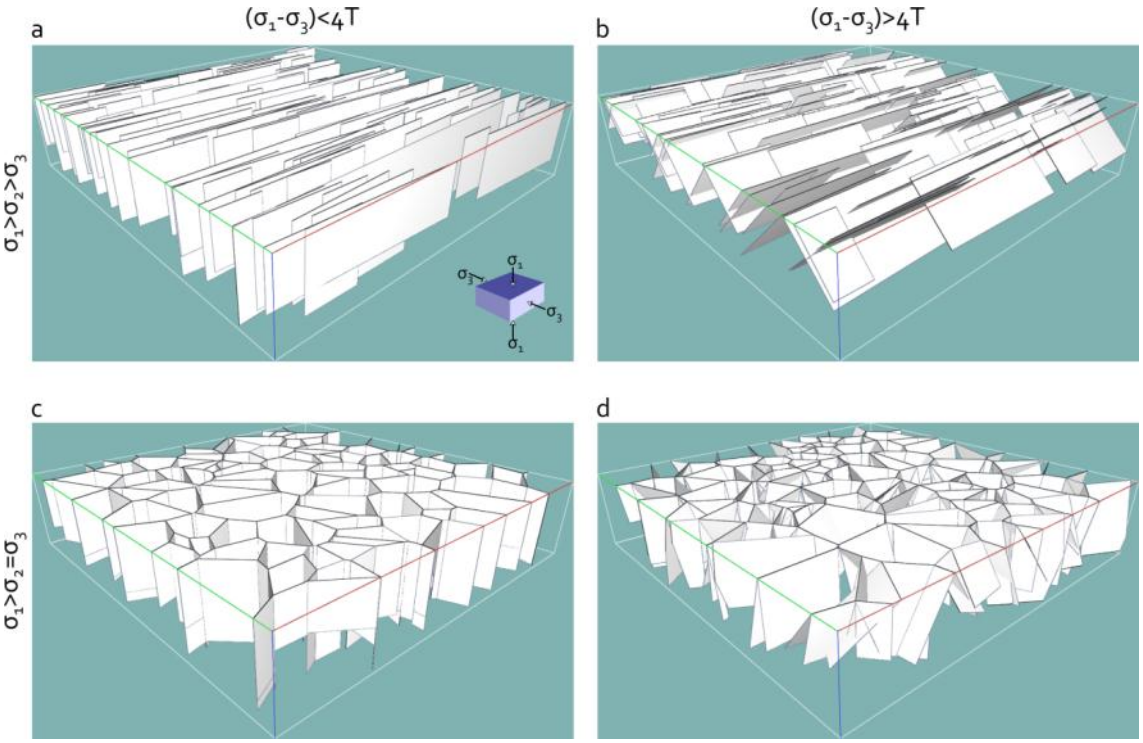


Figure 6.25: Various patterns of fracture networks in three dimensions due to extensional failure criteria.

6.15 Comparison of density measures

A comparison of various density measures is shown in Figure **6.26**, which includes fracture centroid density (FCD, Sect. 6.1), intersection density (ID, Sect. 6.13.2), extended intersection density (EID, Sect. 6.13.3), block centroid density (BCD, Sect. 6.12), backbone density (BBD, Sect. 6.11), and normal intersection density (NID, Sect. 6.13.5). Figure **6.26** also shows the results of fracture cluster, convex-hull and popularity analysis (see definition in the next section). The correlation matrix between the density measures is also given. Visual inspection and interpreting the correlations between density measures suggest that BBD and BCD are highly correlated with ID. The justification of the strong relationship between BBD and BCD is that both measures are direct products of the intersections between fractures in the network. An interesting finding is that by determining ID one can find an approximation of the density of blocks being created due to the extension of existing fractures. This is useful in those applications that are mostly focused on rock mass blocks, e.g., tunnelling in a fractured zone rather than fractures. In addition, FCD shows high correlation with EID and NID suggesting its usefulness as an approximation for them. Note that both FCD and ID can be quickly calculated for any fracture network. Whilst FCD is general, ID requires intersections between fractures; the latter assesses the potential of the network for fluid flow through fractures which is important.

6.16 Popularity index

A measure of popularity of fractures in the fracture network called *popularity index* is proposed as the number of intersections for each fracture in the network. Thus a larger number of intersecting fractures yields a higher popularity measure for fractures. The map in Figure **6.26** is made by discretising fractures with a fine resolution in which the values for cells are the accumulated popularity values. A quick inspection of the results in Figure **6.26** also suggests that the popularity measure is strongly correlated with ID.

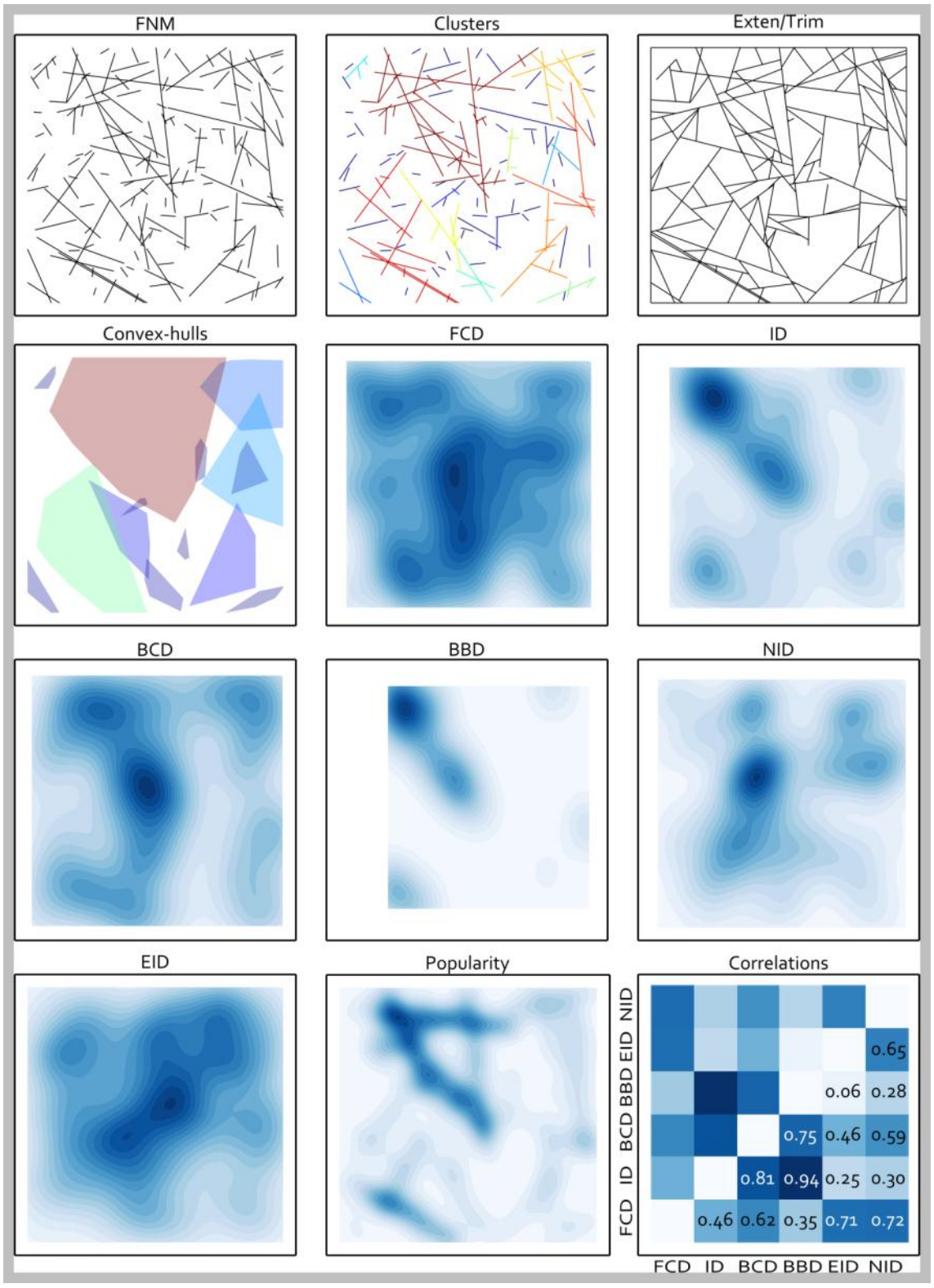


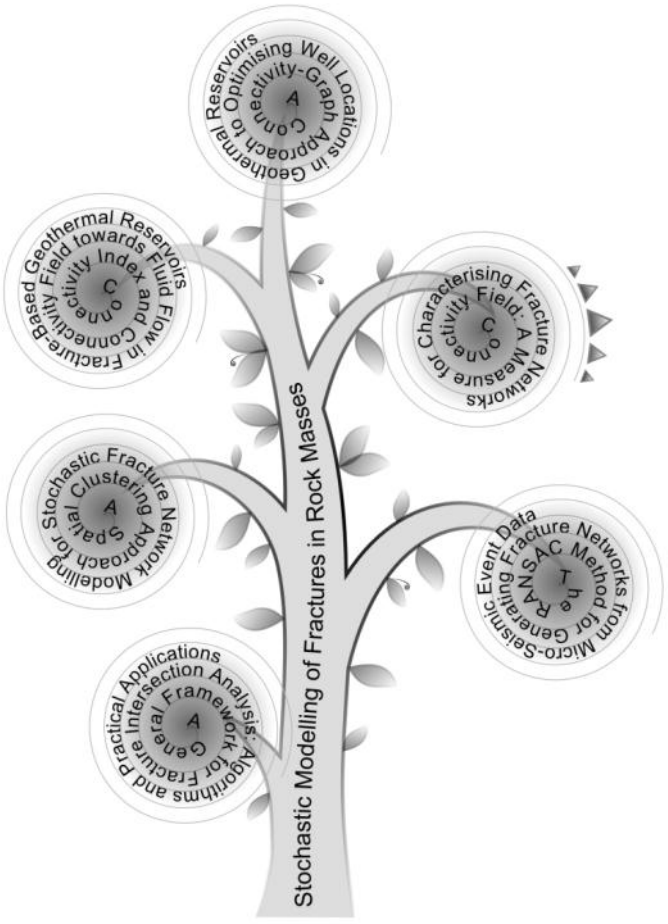
Figure 6.26: Comparison and correlation between various proposed measures. Correlation values are Pearson-r computed pixel-wise.

6.17 Connectivity of fracture networks

For any fracture network that is not completely isolated, the inter-connections between fractures generate fracture clusters (subsets or sub-network of fractures). The connectivity of a fracture network is intrinsically associated with fracture intersections and so with fracture clusters. The connectivity is a topological feature of the fracture network in the sense of spatial association between a fracture and neighbouring fractures. The traditional way of assessing the connectivity of fracture networks is to apply discretisation followed by examination of the connection properties between neighbouring cells in the discretised (*pixelated*) map. This has been reported in the literature for some time (Renard and Allard 2011) and is closely associated with percolation theory. Appling discretisation is straightforward and the resulting maps are binary maps. For a binary map finding neighbouring connected cells is trivial and quick. This suggests that pixel map connectivity assessment is very efficient. However, there are serious drawbacks in the use of discretisation for fracture network connectivity analysis. One fundamental issue is that during the discretisation the intersection (*inter-connection*) information between fractures on the local scale is lost, which may result in highly biased estimation of connectivity, depending to the discretisation setting. The size of pixel (discretised cell) is very much larger than the aperture of fractures (which appears as the width of fracture trace line in two-dimensions). Therefore the local connectivity cannot be reflected (preserved) in the resulting discretised map. Implementing a very fine cell size i.e., close to aperture size to cover the entire fracture network is computationally impractical due to the very large computational power (CPU), memory and processing time. Moreover at such a resolution there will be no additional information other than the fracture network itself. A proper solution would be a method that retains all local and regional inter-connection information for reasonable size of sampling cell, applicable to two- and three-dimensional fracture networks. Chapter 7 comprises a journal paper (Fadakar-A et al. 2014) which proposes, demonstrates and discusses in detail a new measure called the *Connectivity Field* that provides a proper connectivity measure for fracture networks that address the issue discussed above.

7

Connectivity Field: A Measure for Characterising Fracture Networks



This chapter is a journal paper on a new proposal, Connectivity Field (CF). The concept of CF is introduced as a measure that quantifies the connectivity relationship between fractures in a fracture network preserving local and global inter-connectivity information. CF block maps are shown to provide more information on spatial relationships including fracture intersections and clustering. The Generalised CF, the Probabilistic CF and the average CF are all demonstrated to have major potential significance in EGS and also potential application in many areas of DFN engineering.

Statement of Authorship

Title of Paper	Connectivity Field: A measure for characterising fracture networks
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Fadakar-A Y, Dowd P.A, Xu C (2014) Connectivity Field: A measure for characterising fracture networks, J Mathematical Geosciences, DOI 10.1007/s11004-014-9520-7

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	proposing and development of the concepts, conducting simulations and data analysis, developiong algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to the development of concepts, supervising development of work, manuscript evaluation and significant editing and rewriting of the manuscript.		
Signature		Date	28/2/2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to the development of concepts, assistance with editing and evaluation of the manuscript.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

Connectivity Field: A measure for characterising fracture networks

Younes Fadakar Alghalandis[1], Peter A. Dowd[2], Chaoshui Xu[2]

Abstract Analysis of the connectivity of a fracture network is an important component of the design, assessment and development of fracture-based reservoirs in geothermal, petroleum and groundwater resource applications. It is a useful means of characterising the flow pathways and the mechanical behaviours of reservoirs. An appropriate practical measure is required for connectivity characterisation because of the extreme complexity of fracture networks. In this paper, we propose the connectivity field (CF), as a useful measure to evaluate the spatial connectivity characteristics of fractures in a fracture network. The CF can be applied on both a particular realisation of a fracture network model (for deterministic evaluation) and on stochastic fracture network models using stochastic modelling and Monte Carlo simulations (Robert and Casella 1999) for probabilistic evaluation with uncertainties). Two extensions are also proposed: the generalised CF, a measure that is independent of support size, and the probabilistic CF. Potential applications of the CF and its extensions are in determining the optimal location of an injection or production well so as to maximise reservoir performance; and in determining potential flow pathways in fracture networks. The average CF map shows strong correlations with the X_f and P21 measures. The relationships between the CF measures, the fracture intersection density and the fracture network connectivity index are also investigated.

Keywords Connectivity Field, Connectivity Index, Discrete Fracture Network, Fractured Reservoir, Intersection Density.

[1] Corresponding Author: Younes Fadakar Alghalandis. School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, 5005, SA, Australia, Phone: +61 8 8313 1570 Fax: +61 8 8313 4359 email: younes.fadakar@adelaide.edu.au and yfa.study@ymail.com

[2] School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, SA, Australia

7.1 Introduction

Fracture network modelling (FNM) is an important component of the design and development of natural energy and resource systems including geothermal and petroleum reservoirs and aquifers (Freeze 1975; CFCFF 1996; Cacas et al. 2001; Nelson 2001; Jing 2003; Hanano 2004; Kvartsberg 2010; Singhal and Gupta 2010; Fadakar-A et al. 2013a; Seifollahi et al. 2013). In general, the productivity of the reservoir depends critically on connections between injection and production wells and the areal extent of the fracture network in the reservoir. This is particularly true in hot dry rock (HDR) geothermal energy systems, as connections between injection and production wells provide pathways for the geothermal flow. It is, therefore, vitally important to understand the connectivity of the fracture network in such reservoirs and the methods proposed in this paper will contribute to achieving this understanding.

Stochastic FNM provide a means of incorporating uncertainty in the generation of fracture network models; see, for example, Gringarten (1997); Hayashi et al. (1999); Hestir and Long (1999); Dowd et al. (2007); Mardia et al. (2007a); Jing and Stephansson (2007a); Xu and Dowd (2010). Stochastic FNM are based on the theory of random processes and can be implemented by various means including point processes (Diggle 1983; Baddeley 2010) combined, for example, with Monte Carlo simulations. The output of a stochastic FNM is a probabilistic realisation of a fracture network in contrast to conventional deterministic approaches, which produce a single "best" output. The rapid development of computers over the past two decades has made it practical to use stochastic FNM to investigate large-scale probabilistic problems such as modelling fractures in the deep rock masses of an enhanced geothermal system (EGS; Willis-Richards and Wallroth 1995; MIT 2010; Xu and Dowd 2010; Fadakar-A et al. 2013a; Seifollahi et al. 2013), in petroleum (oil/gas) reservoirs (Cacas et al. 2001; Nelson 2001) and in water resource engineering (Freeze 1975; Singhal and Gupta 2010). Stochastic modelling can be applied to many natural resource problems. For example, in geothermal reservoirs, it can be used to simulate fluid flow (Zhang 2002; Fadakar-A et al. 2013b) through fractures (Chilès and de-Marsily 1993; CFCFF 1996; Karvounis and Jenny 2011;

Ghaffari et al. 2012) to assess the main orientations (Xu et al. 2006), preferred pathways and reservoir extents (Fadakar-A et al. 2013b). Discrete Fracture Network (DFN) modelling (Jing 2003), as one implementation of stochastic FNM, represents fractures as individual conducting features (Priest 1993), each having its own physical, mechanical and hydraulic properties. DFN modelling can be done by means of marked point processes (Wen and Sinding-Larsen 1997; Baddeley 2010).

Recent developments in the connectivity assessment of stochastic FNM (see reviews by Michaelides and Chappell 2009; Renard and Allard 2011) include the work by Pardo-Igúzquiza and Dowd (2003) which is basically a cell (pixel) -based connectivity evaluation followed by Xu et al. (2006) who proposed the concept of Connectivity Index (CI) by adapting the connectivity measure so that the connection occurs only through fractures (Figure **7.1**). That is, two sample supports (a *support* is any defined sub-space of the study region varying in size from a point to the entire region) even though they may overlap each other will be connected if, and only if, there is a pathway between them via fractures. This definition of fracture connectivity differs fundamentally from the conventional lattice, pixel or cell-based connectivity measuring methods (e.g., 4 or 8 connectivity scenarios for square cells) in which fractures are discretised into cells or edges and then the relationships between cells (edges) are quantified. In these methods, because of the limitations of the discretisation of the fracture network (e.g., in practice, cells are extremely large compared to the aperture of fractures), the relationships between fractures (intersection, closeness etc.) cannot be fully preserved; in other words, the process of discretisation degrades the physical structure of the connectivity (Figure **7.1**). The CI, on the other hand, investigates directly the probability of connection between two supports via fractures. Figure **7.1** demonstrates some typical situations for which the resulting connectivity differs from lattice-based to support-based methods. In support-based methods the connectivity between two supports is defined only by fracture connections and not by the relative locations of the supports (e.g., overlapping, adjoining), that is, any inter-connection information between fractures is preserved. Consequently, two adjoining supports, e.g., square cells, as in Figure **7.1**(right), are connected if and only if there is a fracture (or set

of fractures) intersecting both. The preservation of connectivity information during sampling (*systematic* or *random* scheme, Figure **7.1**) is fundamental in our research. In Xu et al. (2006) the CI was used to predict preferential flow directions and the output was consistent with that derived by finite element analysis. This type of approach is useful as it partially addresses (by means of stochastic methods) the issue of uncertainty due to lack of data and/or data errors and provides a local interpretation of the entire system behaviour with associated confidence levels.

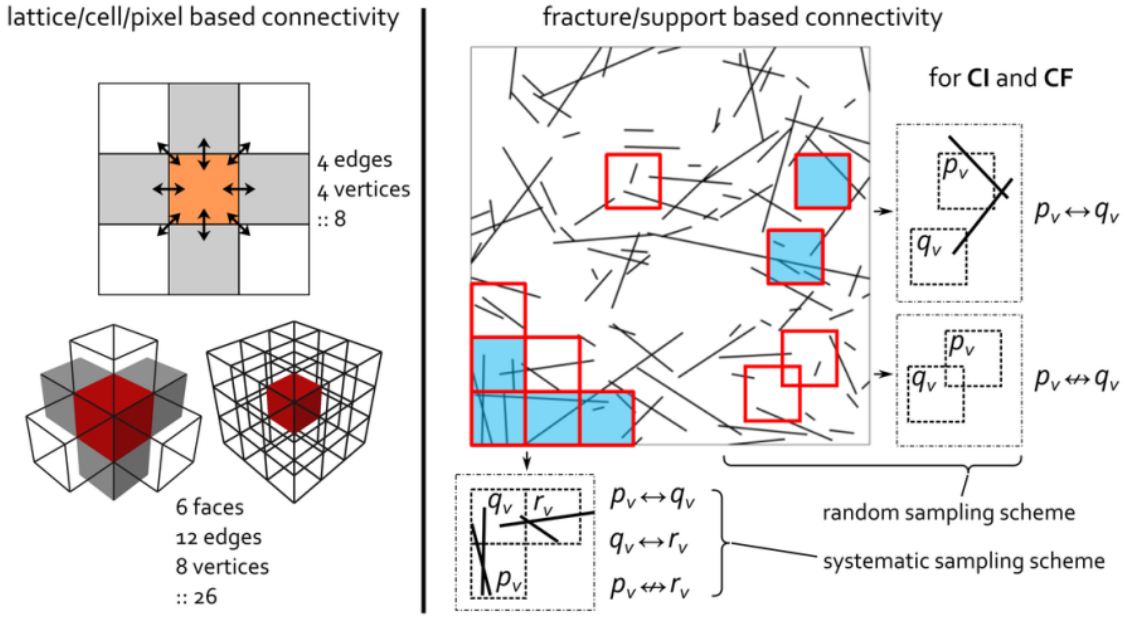


Figure 7.1: Lattice-based connectivity (left) vs. fracture/support-based connectivity (right) measurement. In lattice-based methods the cell is the key element for defining connectivity in three forms: vertices, edges or faces. In support-based methods, however, connection is determined only via fractures. In the example shown (systematic sampling scheme), $p_v \leftrightarrow q_v$ and $q_v \leftrightarrow r_v$ but $p_v \nleftrightarrow r_v$. Also note that two disjoint supports can be connected via fracture(s). Two overlapping supports are not connected if there is no connecting fracture(s).

As noted in Fadakar-A et al. (2011), if the location of each fracture is represented by its centroid, the connectivity of a fracture network (as discussed above) does not necessarily depend on the density of fracture centroids (DFC, see Figure **7.2** for some examples), although a high DFC (and P21) may be associated with a high probability of fracture intersections (Gringarten 1997; see also Mauldon 1992 and 1994; Jimenez-Rodriguez and Sitar 2008, for relationships between P21 (P32) and intersection intensity). For example, areas with a large

number of small, isolated fractures, will have a high DFC (and P21, P32), however, because the fractures are isolated, the associated connectivity value is zero. Three examples of fully isolated fracture networks with associated DFC maps are shown in Figure **7.2**. Despite their DFC patterns all of them have zero intersection densities (void maps). Thus, a more useful and generic connectivity solution would be a measure that counts intersections, and hence inter-connections, between fractures. The general concept of a measure of fracture intersection density has been proposed by other authors. Robinson (1983) in particular, and within the context of percolation theory, proposed a 2D measure as the average number of intersections per line (fracture) at percolation and termed this measure the Critical Intersection Number; other examples include the lineament intersection density in Elfouly (2000). In Fadakar-A et al. (2011) we proposed a different measure, unrelated to percolation theory, in which points (2D) or lines (3D) of intersections are used to calculate the density value per unit area or volume respectively. We termed this measure the (fracture) intersection density (ID). For two-dimensional cases the intersections between fractures always produce points while for three-dimensional cases the resulting intersections can be points or lines (see classifications in Fadakar-A et al. 2011). Furthermore, the resulting intersection lines can also be represented by their midpoints. For both two- and three-dimensional fracture networks a simple density estimation method, such as the Kernel Density Estimator (KDE, with Gaussian kernel, for example), can be used to generate intersection density (ID) maps. The relationship between ID and our present proposal, connectivity field, is also discussed in this paper.

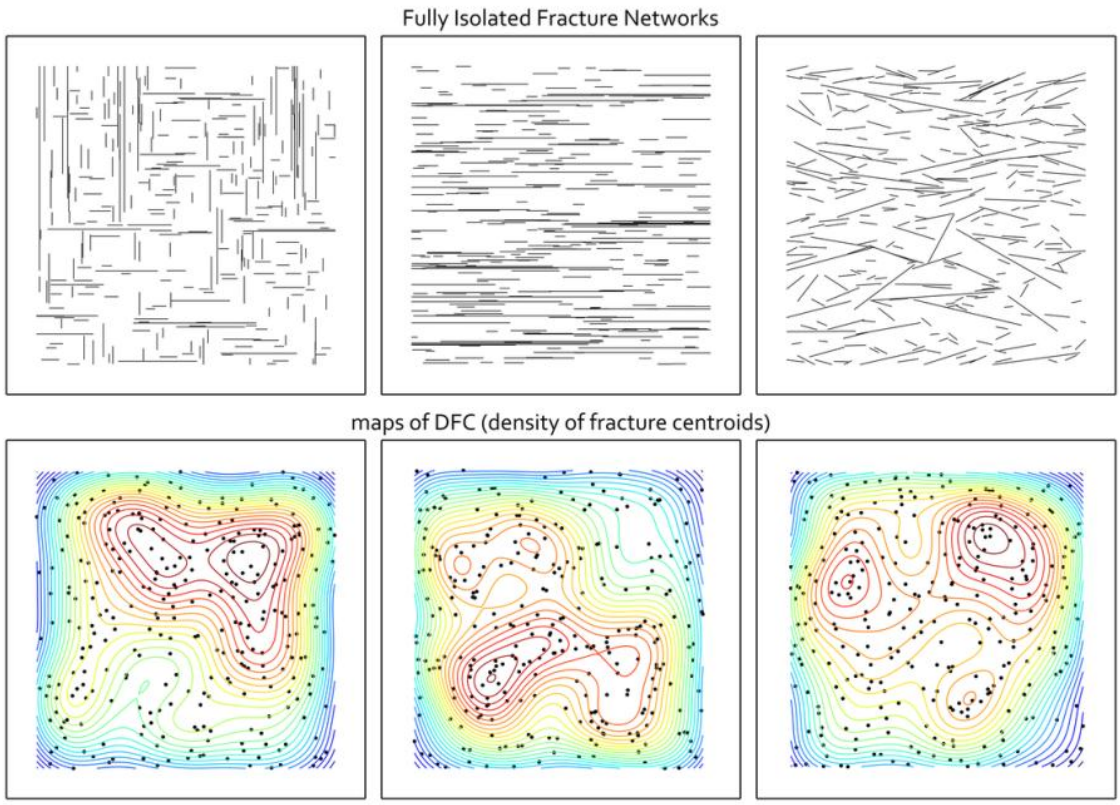


Figure 7.2: Three scenarios for fully isolated fracture networks. As they are fully isolated the ID maps will be void while associated DFC maps are shown above. DFC contours are generated using the KDE method with automatic optimum bandwidth selection.

Two traditional overall fracture network measures are X_f and P21 (P32 for three-dimension, Lee et al. 2010), which measure the average number of intersections per fracture (overall connectivity) and the average fracture length per unit area (i.e., fracture density), respectively. However, both measures consist of a single number for the network, which limits the extent to which they can characterise the FNM in any detail. We propose a series of new concepts to quantify the connectivity of fracture networks in detail both locally and globally. The concepts proposed include the connectivity field (CF), which characterises the spatial connectivity of fracture networks, the generalised connectivity field (GCF), which enhances the CF by eliminating the support dependency and the probabilistic connectivity field (PCF), which produces a connectivity map for a fracture network model. We first define the concepts and then use examples to demonstrate the implementation and to evaluate the performance of the connectivity measures. We also discuss relationships within CF groups and

between them together with other useful measures of fracture networks including DFC, ID, X_f and P21. Potential applications and ideas for further developments are then discussed. Demonstrations are mostly on two-dimensional fracture networks to provide greater clarity in demonstration. Nevertheless, the principles of the methods proposed remain unchanged in their extension to three-dimensional cases. The only additional complexity in three dimensions would be in the computational geometry. An example of CF for a real three-dimensional fracture network is also presented.

7.2 The Connectivity Field

A connectivity function describes the connection between two points (Allard 1993) and can be represented as an indicator function (i.e., 0 or 1). If the two points are connected the function value is 1 otherwise it is 0. This basic concept has been widely applied to evaluate connectivity on the basis of a domain that has been discretised into pixels (cells). Cell-based methods (see the recent review by Renard and Allard 2011) are examples of so-called lattice-based connectivity evaluations. For fracture networks, these methods degrade the connectivity into finite cells. Indeed, by using cell-based discrete representations of fractures it is the relationships between cells that are investigated rather than those between fractures. As a result, the local and global interconnection information among fractures is lost (see example shown in Figure **7.5**). To address these difficulties we have adapted the Xu et al. (2006) definition of the connectivity between two fractures in a fracture network. Fundamentally, if two fractures are directly connected to each other (i.e., they intersect) or are indirectly connected (i.e., there is a pathway via other connected fractures from one to the other) then they have a connectivity indicator of 1. We use $f_1 \leftrightarrow f_2$ to denote that fractures f_1 and f_2 are connected and $f_1 \nleftrightarrow f_2$ to denote that the two are not connected A connectivity measure, such as a connectivity index (Xu et al. 2006) for fracture networks, is defined on the basis of a support (v, sub-space), which is a virtual area (two-dimensional case) or volume (three-dimensional case) of interest in the particular application. The dimension of the support can be as small as zero i.e., a point. If a regular grid is used to cover the entire field of study, say for a systematic sampling

scheme (Figure **7.1**), then the connectivity is defined between two supports (with size v positioned at points p and q) that are either connected via fractures, i.e., $p_v \leftrightarrow q_v$ or are not connected via fractures, i.e., $p_v \nleftrightarrow q_v$. Hereafter we use "*connectivity*" to mean the connectivity between two supports in the sense just defined. Recall that the supports are not physical entities, and thus two nearby or overlapping supports are connected only if there is a pathway from one to the other strictly via fractures (see examples in Figure **7.1**). Note also that if a grid is used to position the supports in a space then the definition of support precludes it being considered as a lattice (e.g., example situations in Figure **7.1**), that is, the relationships between cells (the basis of discretisation methods) are not of interest on their own. For these reasons lattice- or pixel-based connectivity evaluations (see review by Renard and Allard 2011) are not suitable for our purpose which is the direct evaluation of fracture network connectivity via fractures (as demonstrated in Figure **7.1**). A simple application of the defined fracture connectivity assessor would be to compute the connectivity index (CI, $\tau(v)$ in Xu et al. 2006), which is defined as a measure of the probability that two support cells are connected via fractures in a fracture network. The CI is useful for estimating the behaviour of the fracture network in terms of flow pathways. Xu et al. (2006) reported that, although the connectivity index does not deal with mechanical properties of flow through fractures, the resulting preferential flow direction is consistent with the output from conventional deterministic methods such as finite element methods. The connectivity index for two supports (p_v, q_v) in the region $\boldsymbol{\mathcal{R}}$ is defined as:

$$\mathbf{CI} = \mathbf{Pr}(p_v \leftrightarrow q_v), \quad \forall p_v, q_v \in \boldsymbol{\mathcal{R}} \tag{7.1}$$

By definition, the highest CI value is 1 (a cell is always connected to itself if it contains a fracture). For surrounding cells the probability decreases exponentially as the separation distance increases; the minimum value of CI is 0, which means the two cells (supports) are always isolated from each other. CI evaluation is based on Monte Carlo simulation, in which a set of independent realisations of the FNM is generated. The evaluation of the proposed connectivity field in its basic form, however, requires only one realisation (equivalent to the deterministic case). Our proposal, the connectivity field (CF), is a measure that quantifies the connectivity

relationship between fractures in a fracture network preserving local and global inter-connectivity information of fractures. The CF evaluates the connectivity between a set of supports (p_v, q_v) covering the entire study region (area or volume for two- and three-dimensional cases, respectively) as follows.

$$\mathbf{CF} = \left\{ \int_{q_v \in \mathcal{R}} \mathbf{1}(p_v \leftrightarrow q_v) dv \,; \qquad p_v \in \mathcal{R} \right\} \tag{7.2}$$

where 1(.) is the indicator function mapping connectivity to binary values (0 and 1).

CF is thus a surface or volume for two- or three-dimensional regions, respectively. The integral in (2) simplifies to a double summation for a two-dimensional grid of size $m \times n$ covering the region $\mathcal{R}$. If required, a normalisation factor (η) can be used to scale the range of CF to [0, 1], for example for a comparative study with CI (a probabilistic measure). Alternatively, the $\boldsymbol{\pi}_{0,1}(.)$ projection operator can be used. The numerical implementation (discrete form) is:

$$\mathbf{CF} = \left\{ \eta \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}\left(p_{v(ij)} \leftrightarrow q_{v(ab)}\right); \qquad a, b \in \mathbf{G} \right\} \tag{7.3}$$

where $(p_v)_{ij}$ refers to the support indexed at (i, j) in the grid (**G**). When scaling is not required $\eta = 1$. The extension to three-dimensional fracture networks is:

$$\mathbf{CF} = \left\{ \eta \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{s} \mathbf{1}\left(p_{v(ijk)} \leftrightarrow q_{v(abc)}\right); \qquad a, b, c \in \mathbf{G} \right\} \tag{7.4}$$

The evaluation of CF is demonstrated in Figure **7.3** in which the CF is evaluated on a coarse two-dimensional grid (11×11 cells) for the simple fracture network shown in Figure **7.3**a. A more detailed example is shown in Figure **7.4** where the CF clearly identifies cells with the largest hyper-cluster of associated fractures in the field (dark red cells). A hyper-cluster is a set of clusters that are connected to the same support. The CF contour map (e.g., Figure **7.3**c or Figure **7.4**c) provides a smoother representation of the CF map, which is a useful means of visually

depicting the trend in the variation of CF values. In our demonstrations we sampled the CF block map at a higher resolution (say, three times finer) to generate higher quality contour maps. Note that the grid is used here for locating the supports in the study region for a systematic sampling scheme. Recall that cells (supports) in the grid are not connected with each other unless via intersecting fractures. The example in Figure **7.5** demonstrates the lattice representation of connectivity for the fracture network shown in Figure **7.4**a. As shown, the lattice-based connectivity evaluation (here edge connection only) generates a single large cluster (which covers almost the entire study region) for the same size of support used in CF. The CF on the other hand provides much more information about the spatial relationships among fractures e.g., intersections and clustering. Lattice-based evaluations cannot provide such information because, after discretisation of the fracture network into the lattice, the original relationships between fractures are degraded to the cell size, whereas in CF, connectivity is still determined by fractures regardless of the support size (as shown in Figure **7.1**). Thus CF characterises fracture networks while preserving local and global relationships among fractures (Figure **7.5**).

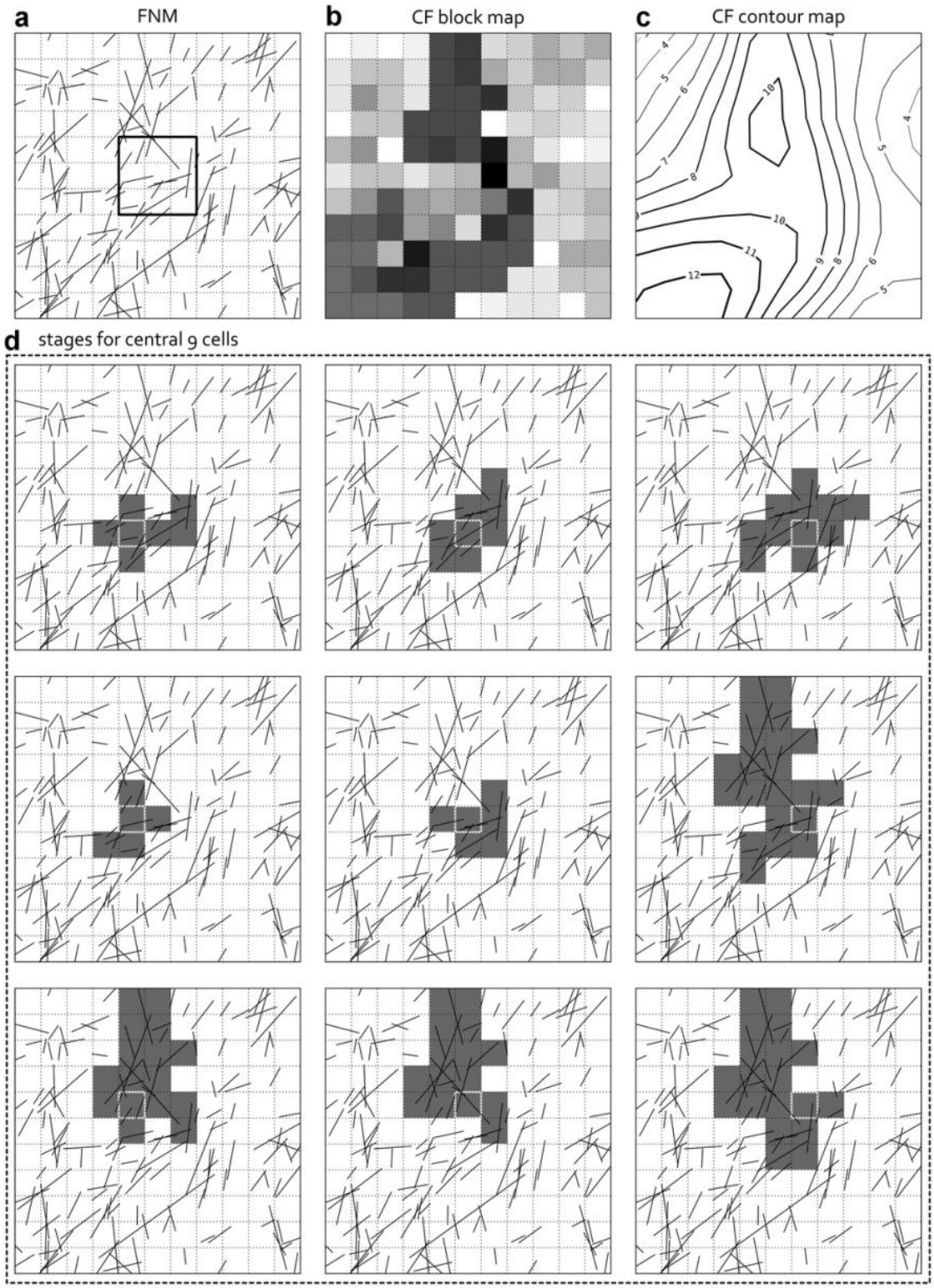


Figure 7.3: The computational stages of the CF; (a): A realisation of a FNM; (b): The resulting CF block map. The darker the colour the higher the value of CF; (c): The smoothed contour map of super-sampled CF. (d): stages for central 9 cells.

As fracture connectivity results from fracture clustering, CF provides a more reliable connectivity assessment of a fracture network because of its intrinsic link to the fracture clusters. In fact, fracture cluster analysis is used in the calculation of the CF in order to associate connected fractures with a support cell. In advanced forms of intersection analysis, the ID measure identifies regions with higher probability of flow through fractures and these correspond to high densities of intersections among fractures in the network (see Figure **7.8**(ID); Fadakar et al. 2011). The CF achieves the same end as ID but provides much more reliable maps because of its direct evaluation of the connectivity in the fracture network (see Figure **7.8** and Sect. 7.3 for detailed comparison).

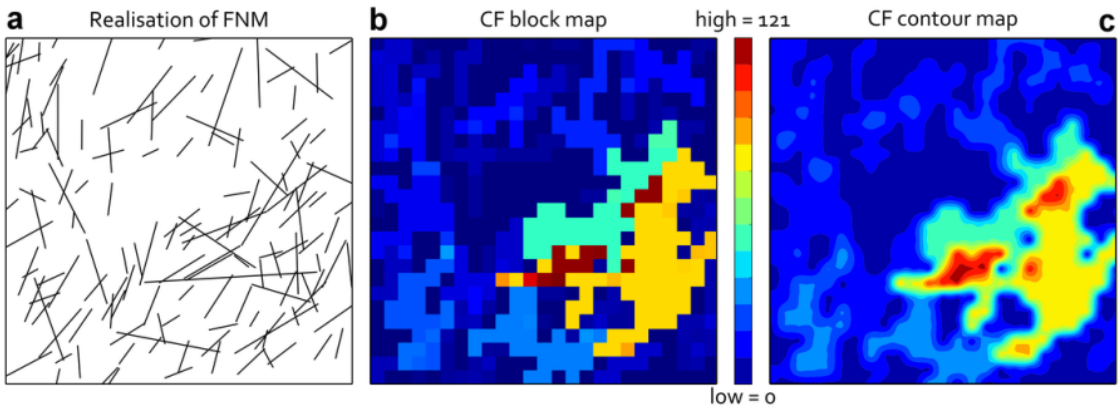


Figure 7.4: A realisation of a FNM, the CF map computed on a 25×25 grid and the interpolated contour map of the CF, from left to right respectively.

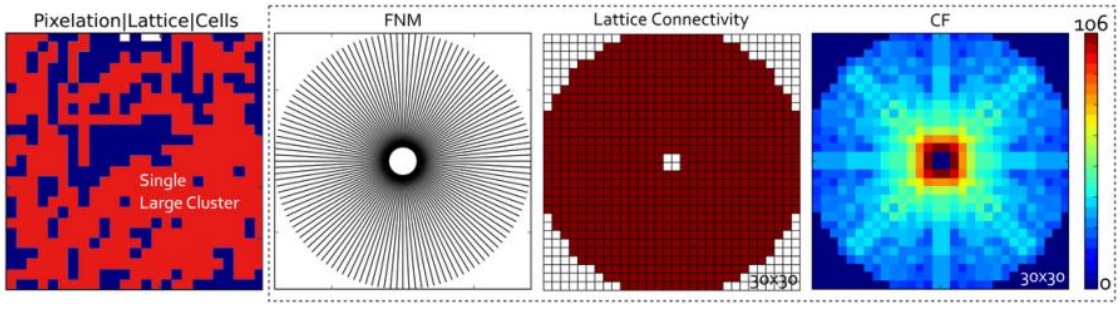


Figure 7.5: Lattice-based connectivity evaluation of a fracture network. For comparison, the example fracture network is the same as that used in Figure **7.4**a for the CF map. Note the difference in the output of lattice-based connectivity and support-based connectivity evaluations on a synthesised ray form fracture network.

7.2.1 The Generalised Connectivity Field

The CF evaluation described above is based on a fixed size support v. It is, therefore, of interest to investigate the sensitivity of CF to variation in the support size (i.e., $v \in \{\varepsilon..\boldsymbol{\mathcal{R}}\}$ where $\boldsymbol{\mathcal{R}}$ is the entire field of study and $\varepsilon \to 0$ i.e., a point support). In the example shown in Figure **7.6**a, we investigate the variation of CF maps for support sizes in the range between 1/9 and 1/400 of the entire area. We observe that the spatial characteristics of the CF remain more or less similar under change of support size. This leads to the proposal of the generalised form of the connectivity field (GCF), which is defined by incorporating a wide range of support sizes (ε to $\boldsymbol{\mathcal{R}}$) and is formulated in continuous form as:

$$\mathbf{GCF} = \eta \int_{\varepsilon}^{\boldsymbol{\mathcal{R}}} \mathbf{CF}\, dv \tag{7.5}$$

Furthermore, the discretised form of the GCF can be calculated as:

$$\mathbf{GCF} = \eta \sum_{i=1}^{n} (\mathbf{CF})_{v=\frac{\boldsymbol{\mathcal{R}}}{i}} \tag{7.6}$$

in which for $i = 1$ the support size becomes equal to the entire region $\boldsymbol{\mathcal{R}}$ (upper bound), and $i = n$ determines the lower bound for the support size. The GCF is therefore an E-type map of all CF maps obtained from a wide range of support sizes (Figure **7.6**a). Figure **7.6**c and 7.6d show an example of the GCF map for the fracture network shown in Figure **7.6**b.

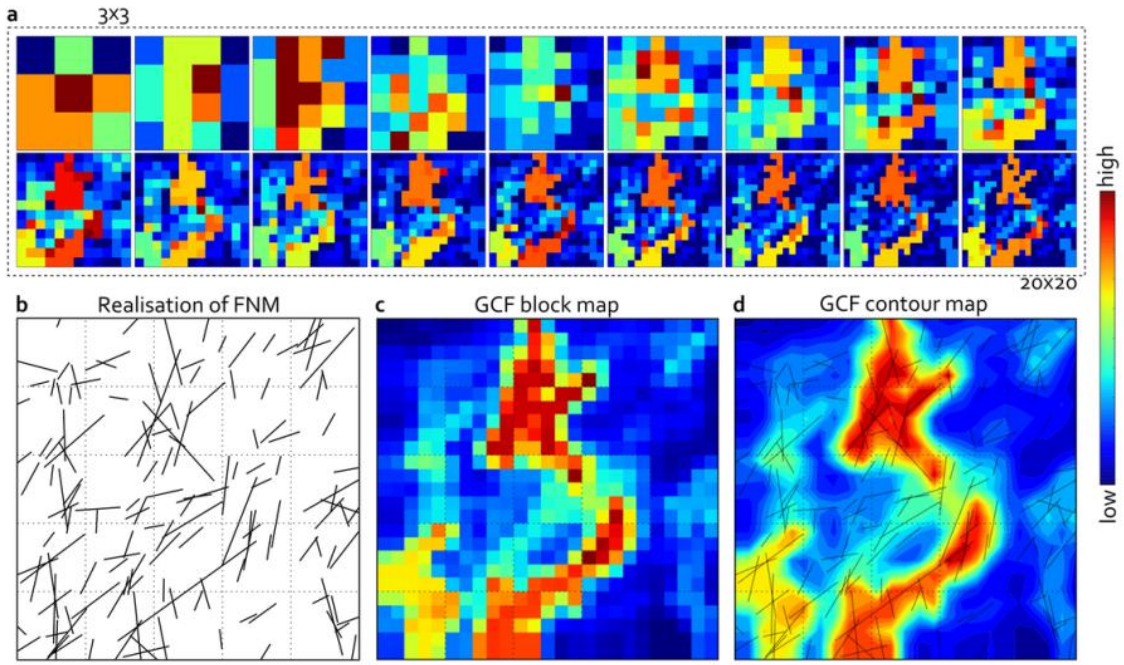


Figure 7.6: An example of the evaluation of GCF; (a): First stage: evaluation of CF for different support sizes v; (b): the fracture network; (c, d): The resulting GCF maps.

The variation in CF values as a function of support size is shown in Figure **7.7** for four realisations. All four realisations have a more or less constant median CF under significant variation in support size. The apparent similarity in the spatial pattern of the CF maps (for various support sizes, Figure **7.6**) and the invariant median statistic indicate that CF is a resistant measure with respect to changes in the support size. More importantly, these observations also suggest that CF is, to some extent, additive (i.e., almost linearly scalable) thereby increasing the reliability of the proposed GCF formulation. Compared to Figure **7.4**, in which the CF contour map is a product of super-sampling (i.e., improving the resolution) of the CF block map, the contour map of GCF is constructed directly on top of the GCF block map. In other words, GCF eliminates the need for super-sampling and so adds further reliability to the connectivity mapping of the FNM by removing possible biases due to the super-sampling.

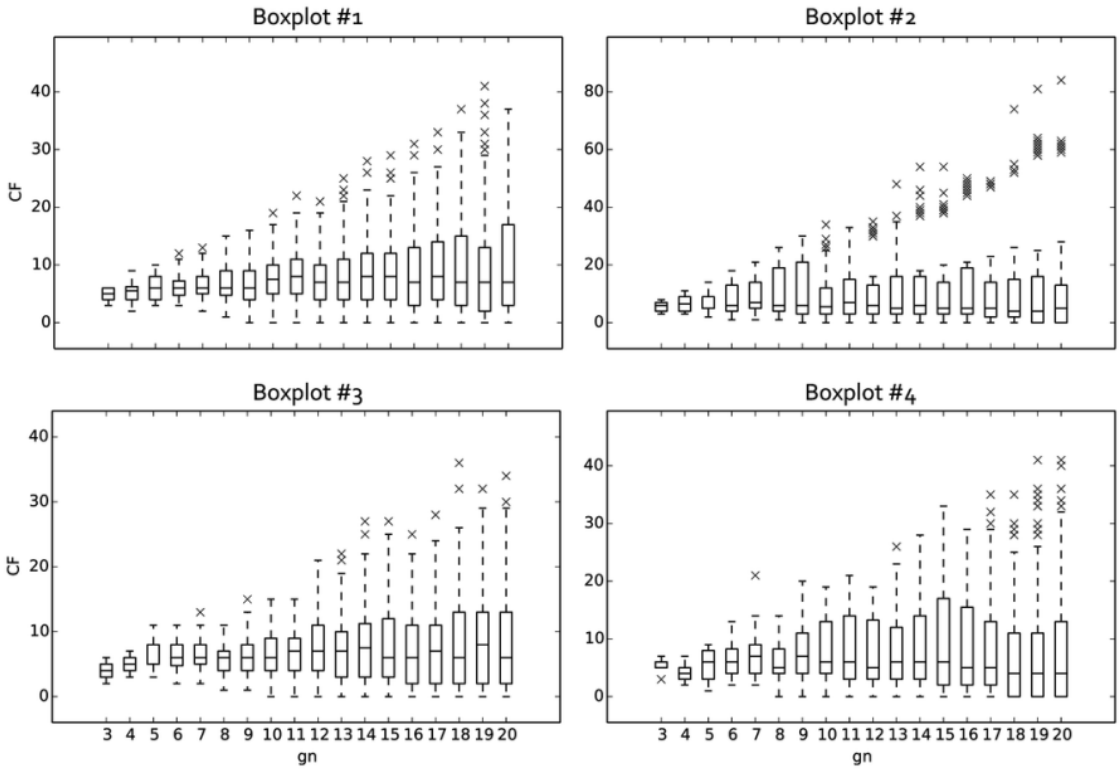


Figure 7.7: Box-plot of CF values for the various support sizes (1/3 to 1/20 of region $\mathcal{R}$) shown in Figure **7.6**a. An interesting observation is that the median CF value for different support sizes is less variable (note that here the CF values are not normalised). Increasing the resolution generates a longer positive tail for the CF distribution.

7.2.2 The Probabilistic Connectivity Field

The CF, as defined above, is calculated from realisations of the FNM. We have extended the definition in order to derive the underlying governing function of the connectivity field. This extension, termed the probabilistic connectivity field (PCF), combines of CF and Monte Carlo simulations to describe the connectivity characteristics of a FNM. It is the application of CF to a stochastic model of the fracture network and is defined as:

$$\mathbf{PCF} = \boldsymbol{\pi}_{0,1}\left(\sum_{i=1}^{n} \mathbf{CF}\right) \tag{7.7}$$

where n is the number of independent realisations of the FNM, and $\pi_{0,1}(.)$ is a projection operator to the range [0, 1] to comply with a probability measure. An example based on 60 realisations of a FNM is shown in Figure **7.8**. In this example,

the realisations were generated using the DFN approach (Jing 2003; Alemanni et al. 2011) with the following settings: the locations of fractures were simulated by a Homogeneous Poisson Point Process (HPPP) with point density of λ=200 (stationary case); the orientations of fractures were derived from a von Mises-Fisher distribution ($\kappa \leq 1$, almost isotropic orientations; Baddeley 2010); fracture lengths were generated from an exponential distribution (Priest and Hudson 1981; La-Pointe and Hudson 1985) with λ=0.1. The resulting PCF in Figure 7.8(PCF) shows the probability that locations are connected to (and hence are part of) fracture clusters. In other words, the higher the PCF value, the more likely the cell is connected to the reservoir model. The CI map in Figure 7.8 shows the connectivity index for the centre point of the realisation and is much more homogeneous than the either the ID or the PCF map. The ID map is an E-Type map of the 60 individual ID realisations and appears more homogeneous, or smoother, than the PCF map, which is the sum of the 60 CF realisations. In a sense ID is a, more or less imperfect, proxy for the real connectivity quantified by CF and this is reflected in the greater homogeneity or smoothness of the former. As PCF quantifies the underlying reservoir connectivity of the FNM it provides a more realistic stochastic model of connectivity.

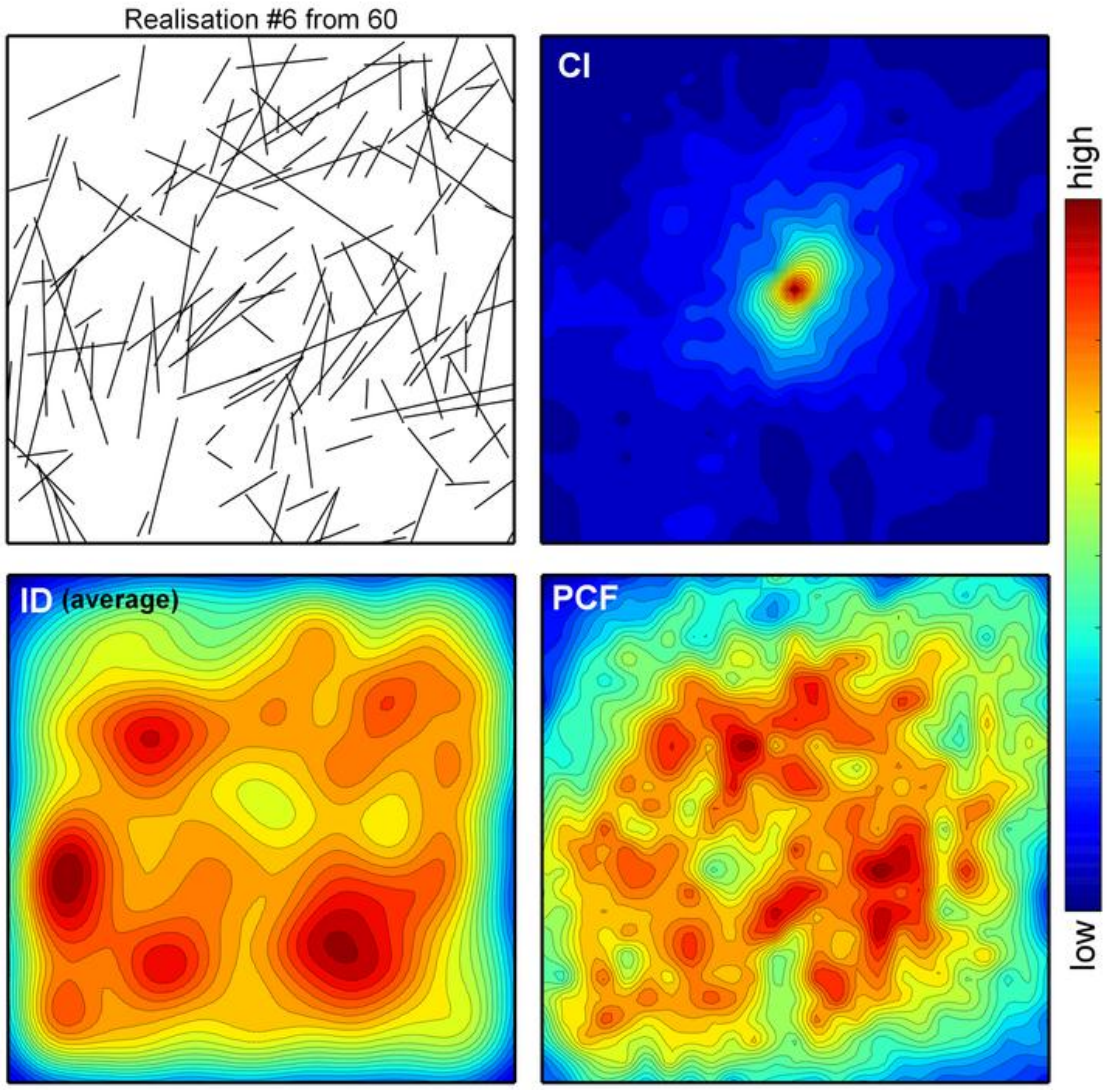


Figure 7.8: Comparison of ID, CI and PCF for a set of 60 realisations of a FNM as input data. One realisation (#6) is shown as an example.

7.3 The relationships between the CF, GCF, DFC, ID, X_f and P21

The intersection density (Gringarten 1997; Elfouly 2000), as developed further in Fadakar et al. (2011), is a measure of the intensity of fracture intersections in a FNM. In a realisation of a two-dimensional FNM, fracture intersection points (used to calculate the ID values) may differ from the points used to represent fracture locations (e.g., homogeneous point pattern; Baddeley 2010) and therefore the ID of a fracture network differs from the corresponding DFC map (see examples in

Figure **7.2**). DFC is a global characteristic of the FNM (i.e., the intensity value of the associated point pattern if a point process is used to describe the FNM) which does not include any relationship among fractures. On the other hand, the ID is a direct measure of intersections between fractures in the FNM and is thus an effective measure for analysing fracture network properties such as percolation state and flow. The ID is, therefore, useful for identifying regions with high densities of fracture intersections, which can be used to help determine optimal well locations in petroleum or geothermal engineering applications (Fadakar et al., 2011). However, the ID provides no information about clustering or connectivity at the reservoir scale. For example, a high fracture intersection area may be isolated and not connected to (and hence not part of) the percolating fracture cluster spanning the reservoir but the ID alone does not provide this connectivity assessment. The proposed CF can be used directly to solve this problem.

Figure **7.9** shows a comparative study of the ID and the CF for four realisations of a FNM. The correlation coefficients between ID and CF for the four realisations are, respectively, 0.26, 0.20, 0.42 and 0.34 and hence there is weak correlation between the ID and the CF. This confirms the fundamental difference between ID and CF in characterising fracture networks. Although ID is much more reliable than the DFC in identifying regions with high fracture intersections (Figure **7.2** and Figure **7.10**), the cluster connectivity of regions at the reservoir scale can only be described by the CF. The two may differ significantly (Figure **7.9**) and it appears that, at the reservoir scale, the actual flow response is more closely related to the CF.

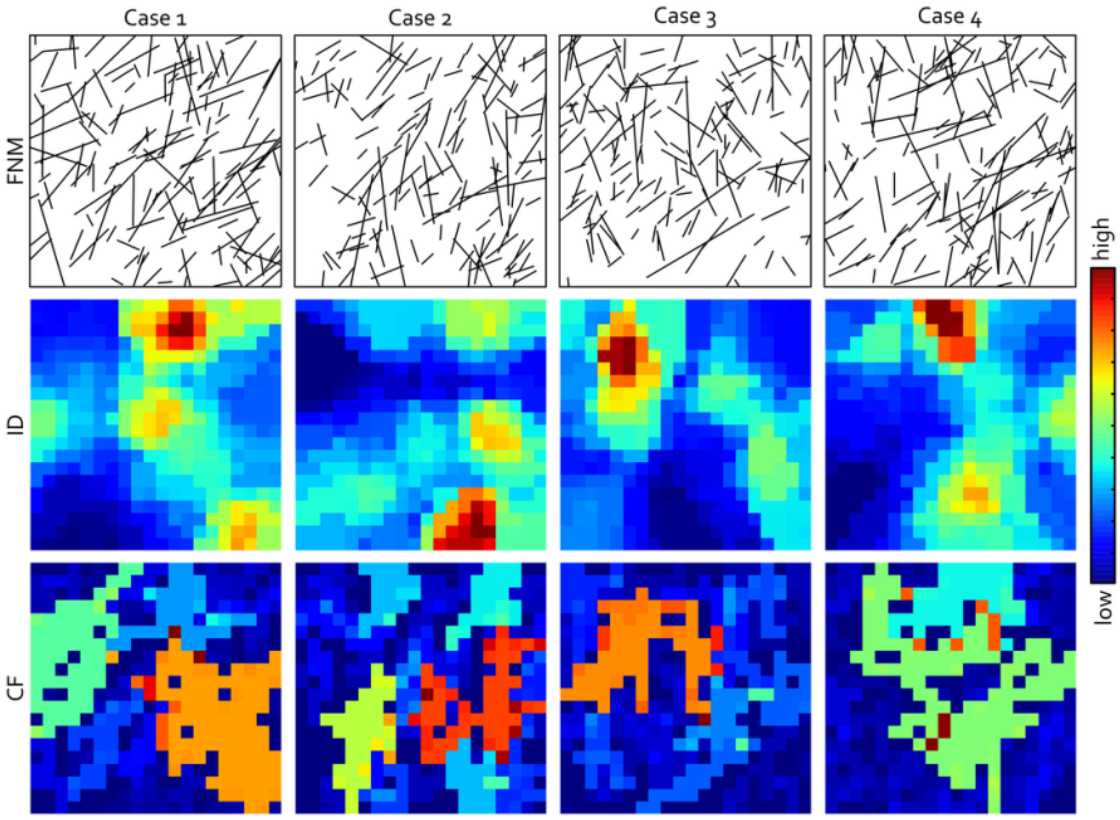


Figure 7.9: Comparison of CF and ID: four realisations (cases) of a FNM and the corresponding ID and CF.

DFC and the P21 and P32 measures are useful conventional ways of analysing the intensity of fractures (see also Zhang and Einstein 2000), assessing point patterns and providing information for modelling. The intensity measures are useful means for probabilistic assessment of rock quality in which the rock mass is the focus rather fractures. The reliability of these measures for flow modelling, however, is questionable (see Figure **7.2** for examples in which fully isolated fracture networks show good DFC maps). Thus a measure such as ID would be more useful for flow applications, because higher intersection density suggests higher possibility of flow. The DFC contour is not an effective way of determining the interacting regions (fracture connection) in the fracture network whereas ID provides a more informative map in terms of fracture network connectivity (see comparison in Figure **7.10**). The drawback of ID is that it does not contain any information about fracture dimensions as can be seen in Figure **7.10**(ID contour), which causes biases in the evaluation of fluid flow pathways. Connected fractures create clusters and pathways for flow can be depicted in fracture clusters. Clusters,

for a given support in CI and CF, create hyper-clusters. CF deals directly with hyper-clusters and it therefore effectively characterises the connectivity of the FNM at the reservoir scale (compare the maps in Figure **7.10**). By definition, GCF and CF are intrinsically related (Eqs. 7.2 and 7.5). The CF is reasonably resistant to changes in the support size whereas GCF does not depend on the support size. CF (and, even more so, GCF) incorporates all density, intersection and clustering information and thus provides maps suitable for determining flow pathways in fracture networks regardless of their pattern and complexity. Further investigations (Figure **7.11**) show that CF and GCF are highly correlated with a correlation coefficient of 0.81 from 60 FNM realisations (this can also be seen visually in Figure **7.10**).

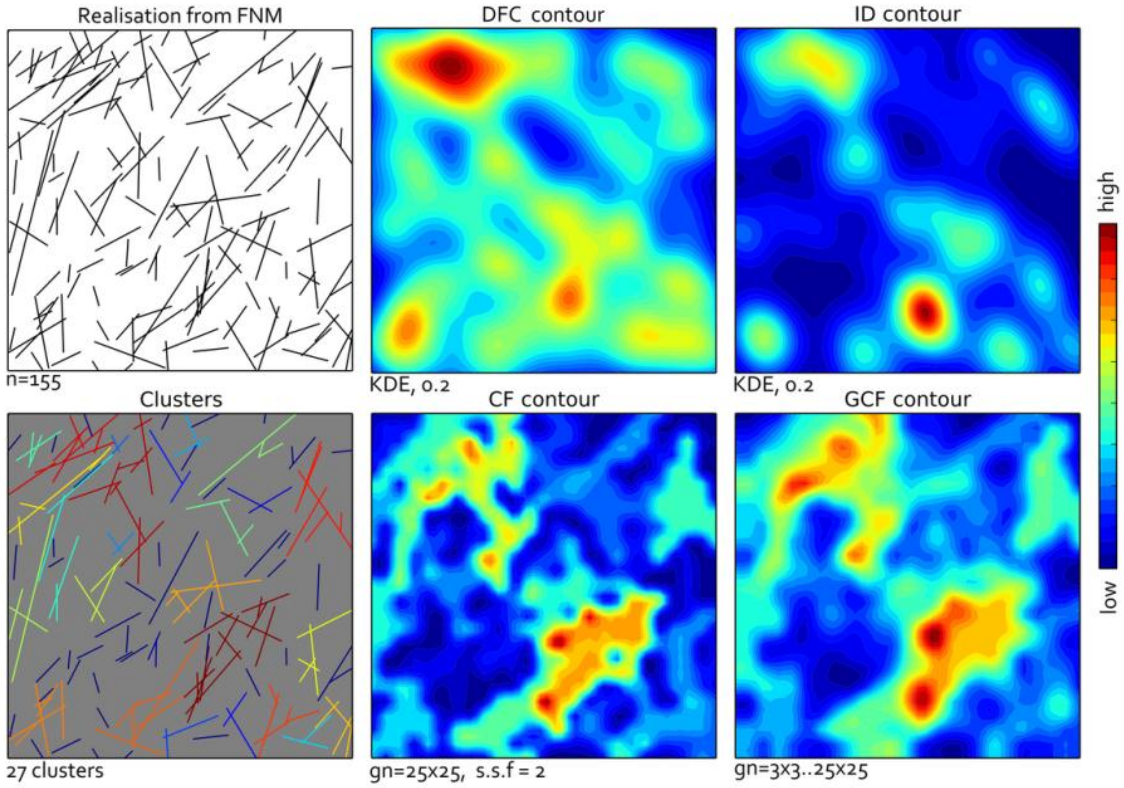


Figure 7.10: Comparison of various FNM measures: DFC, ID, CF, GCF and fracture clusters.

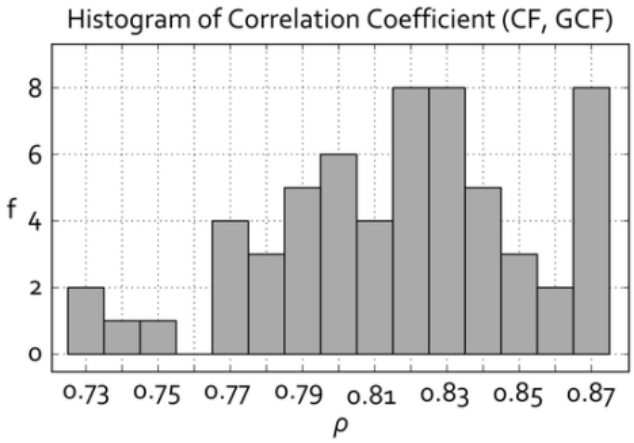


Figure 7.11: Correlation coefficient between CF and GCF for 60 realisations of a FNM; While support size for CF was 0.16% of the size of study region $\mathcal{R}$, for GCF it varied from 0.16% to 100% of $\mathcal{R}$. CF and GCF are highly correlated with an average correlation coefficient of 0.81.

Figure **7.12** shows relationships between two conventional measures, X_f and P21, and the average connectivity field ($\overline{\mathbf{CF}}$, a scalar), defined as:

$$\overline{\mathbf{CF}} = \frac{\sum \mathbf{CF}}{\#(\mathbf{CF})} \tag{7.8}$$

where #(.) is the number of elements in the CF for grid-based sampling. The investigation was conducted using 700 realisations of a FNM (see details in the caption of Figure **7.12**). The curves (solid red) were fitted using polynomials of order 4. The example shown in Figure **7.12** is one of many simulations in which nonlinear relationships between CF and X_f and P21 are demonstrated. While correlations between pairs are positive and strong, the apparent S-shape behaviours between X_f and $\overline{\mathbf{CF}}$, and P21 and $\overline{\mathbf{CF}}$ are interesting. The S-shape reflects a *phase transition* in the phenomenon under study. For connectivity in fracture networks it reflects the transitional increase in CF values corresponding to values of the measure greater than ~0.5. For P21 versus CF the transition to higher CF values occurs at around P21=0.6. From this we can conclude that $\overline{\mathbf{CF}}$ (as a single scalar form of CF) is also a potentially effective measure for characterising fracture networks from P21 and X_f perspectives.

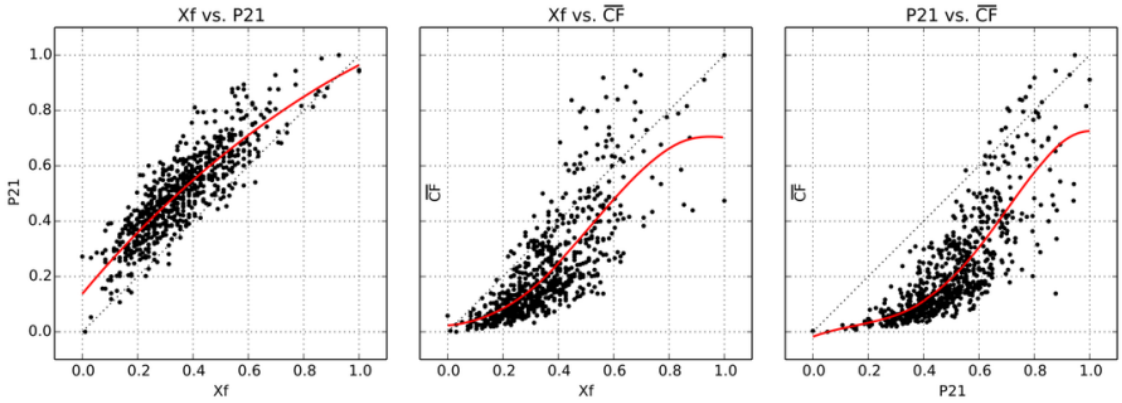


Figure 7.12: Scatter plots for X_f, P21 and $\overline{\boldsymbol{CF}}$ measures based on 700 realisations from FNM (locations: Homogeneous Poisson Point Process with $\lambda = 100$; orientations: von-Mises with mean direction on X axis and $\kappa = 0$; lengths: Truncated-Power-Law [0.03, 1]). Note that all measures are normalised to the range [0, 1]. Fitted curves are polynomials of order 4. Support size for $\overline{\boldsymbol{CF}}$ was $\boldsymbol{\mathcal{R}}/625$.

7.4 Applications of the Connectivity Field group

7.4.1 CF and Flow Pathways

The CF can be used to determine existing and possible flow pathways through fracture networks. Figure **7.13** demonstrates clearly the two potential pathways (dashed and dotted lines) in the network based on the CF output. The pathway lines are constructed by joining high CF-value cells in the CF map (see Figure **7.14**). The CF map can also help identify those pathways that could be constructed under minimum development (e.g., fracture growth) of the fracture system (regions $\Re_1$ and $\Re_2$). This result, when associated with the geo-mechanical properties of the fracture network (such as the stress field), provides an even more accurate model. Such information is useful in determining the potential connections in the fracture network, for example as a result of a fracture stimulation process in geothermal reservoir applications.

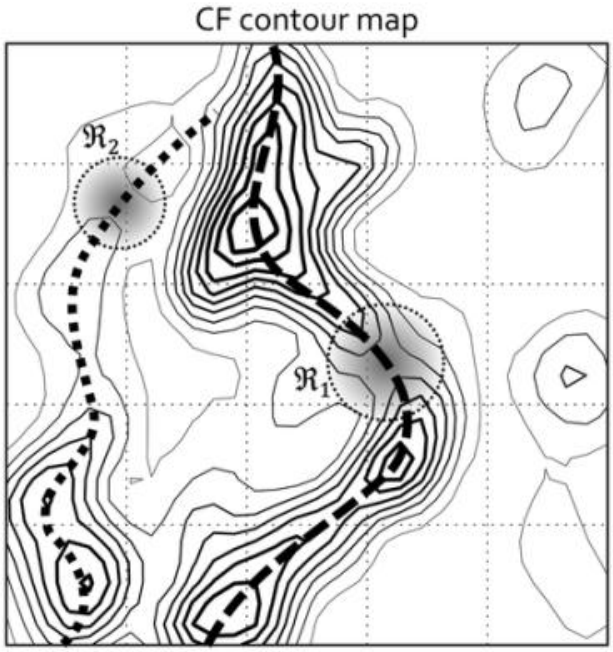


Figure 7.13: Pathway analysis using the CF for the fracture network shown in Figure **7.6**b. Two regions, $\Re_1$ and $\Re_2$, are also identified as having the highest potential to change the percolation state of the region (see next section). The dashed line represents the major flow pathway while the dotted line shows a less developed pathway from the bottom to the top of the region.

7.4.2 CF and Percolation State

The percolation state (Stauffer and Aharony 1992; Robinson 1983) is used to describe a condition in which the reservoir is permeable because the internal structure has reached a state in which two sides (or, in general, *n*-sides) of the region of study are connected and hence the internal structure of the region is permeable (Sahimi 1993). In our application, this internal structure consists of connected fracture clusters; and the reservoir percolation state is reached only if at least one pathway has been established connecting the input and the output sides of the region through fracture clusters. Although there are many publications on the use of percolation theory for lattice-based connectivity assessments of fracture networks (e.g., Hoshen and Kopelman 1976; Balberg 1986; Berkowitz 1995; Renard and Allard 2011), our aim here is to investigate the relationship between fracture connectivity, as defined in this paper, and the percolation state. We consider each side of the region as an individual domain (support) and use the term multi-cluster to denote a set of fracture clusters intercepting a side. Reaching the percolation

state is, therefore, essentially equivalent to the connectivity between multi-clusters as a system response to changes in the fracture network. By definition, the percolation state is related to connectivity measures. For example, the relationship between the percolation state and the CI is reported in Xu et al. (2006) as:

$$\mathbf{P}_{state}(S_1, S_2) = \mathbf{Pr}\left(\bigcup_{p_v \in S_1} \bigcup_{q_v \in S_2} (p_v \leftrightarrow q_v)\right) \leq \sum_{p_v \in S_1} \sum_{q_v \in S_2} (\mathbf{CI})_{p_v, q_v} \tag{7.9}$$

That is, the upper bound of the percolation state can be achieved by summation of all CI of point pairs (p_v, q_v) from sides S_1 and S_2. It is important to note that CI maps are always probabilistic and so the relationship in (9) is an evaluation of the chance of reaching the percolation state for fracture network models. This implies that for a given deterministic fracture network (i.e., one particular realisation), since the CI is not defined the relationship in (9) cannot be used, although the percolation state of the fracture network can be calculated.

By definition, the CF is directly related to the connectivity of the hyper-clusters in the fracture network and hence it uses fracture-clustering information and so provides the possibility of evaluating the percolation state of an FNM. In the use of CF for analysing the percolation state the potential of each cell in the grid of the network to trigger the state change is quantified. As demonstrated in Figure **7.14** the highest values of the CF are usually obtained where there are small gaps (relative to the size of the support) between clusters of fractures. Connecting these high CF values in the CF map identifies the fully or partially developed pathways in the network. In Figure **7.13** the regions ($\Re_1$ and $\Re_2$) with slightly lower values between the summits are the areas with the highest potential to change the state of the system, i.e., to reach the percolation state. This demonstration indicates a potential use of the CF in *predicting* a state in which percolation in the fracture network may occur. The procedure can be applied in a Monte Carlo simulation to estimate the percolation state for fracture network models.

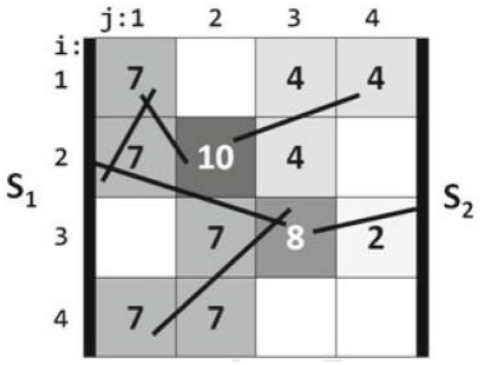


Figure 7.14: A system of fractures that is percolating for sides S_1 and S_2; the numbers shown are the CF values for each cell. The cells with higher CF values i.e., $cell_{(2,2)}$ and $cell_{(3,3)}$ are used to assess the percolation state of the system.

7.4.3 Using CF to determine well locations and to design underground repositories

Another application of CF analysis is in determining optimal drilling locations for geothermal, petroleum or groundwater reservoirs to maximise the connectivity between a proposed new well and the reservoir (see the example demonstrated in Sect. 7.5) and hence contribute to maximising reservoir performance. CF analysis can also be applied to the design of underground repositories for hazardous wastes or for carbon dioxide sequestration, where it is required to determine a location that is not well-connected to the surrounding environment. For these applications, areas with minimum CF values are the most suitable regions. A final note is that, incorporating variable permeability for fractures would increase the reliability and effectiveness of the output; this will be the subject of future work (e.g., a proposal for weighted connectivity as in Fadakar-A et al. 2013b).

7.5 CF applied to a real three-dimensional fracture network

The definition of CF and its extensions are generic and hence equally applicable to both two- and three-dimensional cases. The evaluation of CF requires efficient computational geometry algorithms and implementations in three dimensions. The intersection analysis between two fractures in three-dimensions results in 0, 1 or many intersection points, i.e., no intersection (0), edge intersection (1) or line

intersection (many). In addition, the support v is a cube (a simplistic form of a three-dimensional cell, see Figure **7.15**). The procedure for evaluating the CF in three-dimensions begins by finding the extent (bounding box: $\mathcal{X}$) of the fracture network, which is then divided into cubic cells (other shapes can also be used) according to the resolution (i.e., number of cells $\{n_x, n_y, n_z\}$ or the dimension of support v) required, see Figure **7.15**a to 7.15d. The CF value for each cell is then calculated. Figure **7.15**e and 7.15f show the resulting three-dimensional CF slice and volumetric maps.

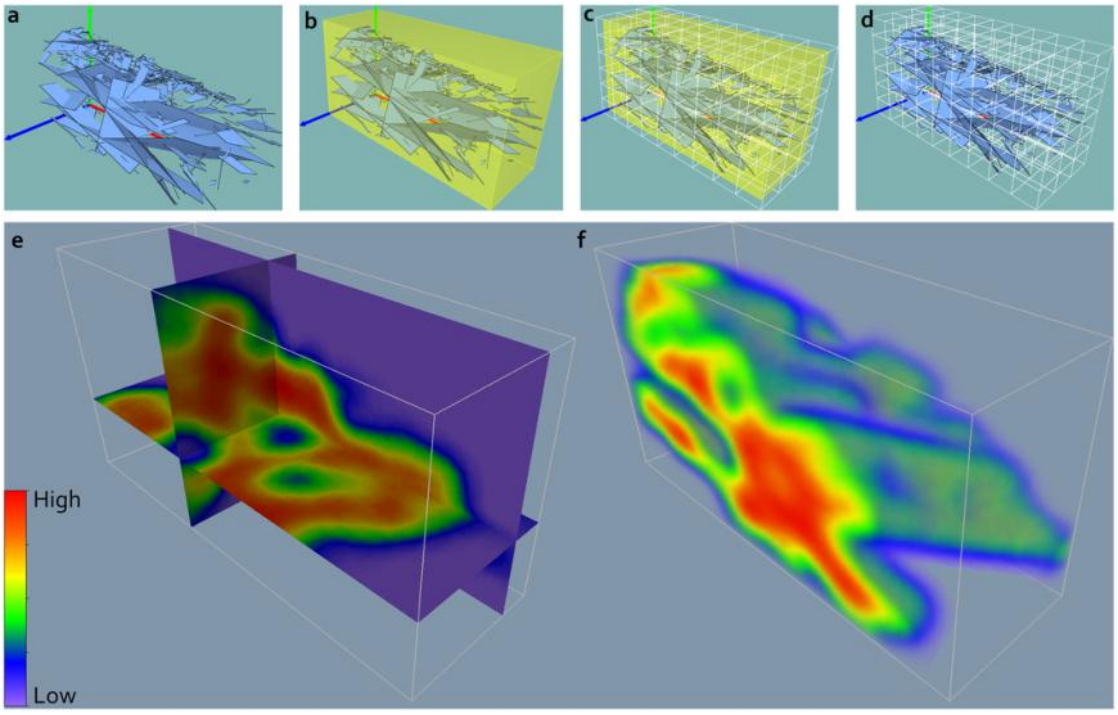


Figure 7.15: The CF for the Leeds Rock Fracture Data Set; (a to d): Stages for the preparation of the three-dimensional grid for CF evaluation; the evaluated CF maps are shown as: (e): the interpolated slice map, (f): the volumetric rendered map.

The case study shown in Figure **7.15** is based on the public domain Leeds fracture data set (Dowd et al., 2009; Leeds 2011), which was constructed from a set of measurements of fracture traces on seven horizontal slices of a block of granite with an extent of approximately $180 \times 90 \times 60$ cm. The CF maps of the block (Figure **7.15**e and 7.15f) clearly demonstrate some well-connected areas in the block (areas shown in red). These areas can also be considered as the controlling domain for characteristics such as fluid flow through the fractures within the block. In addition, locations with high CF values are potentially the most suitable drilling

locations for wells with good connections to the reservoir extent. The CF map also indicates the overall shape of the connected reservoir based on the existing fracture network.

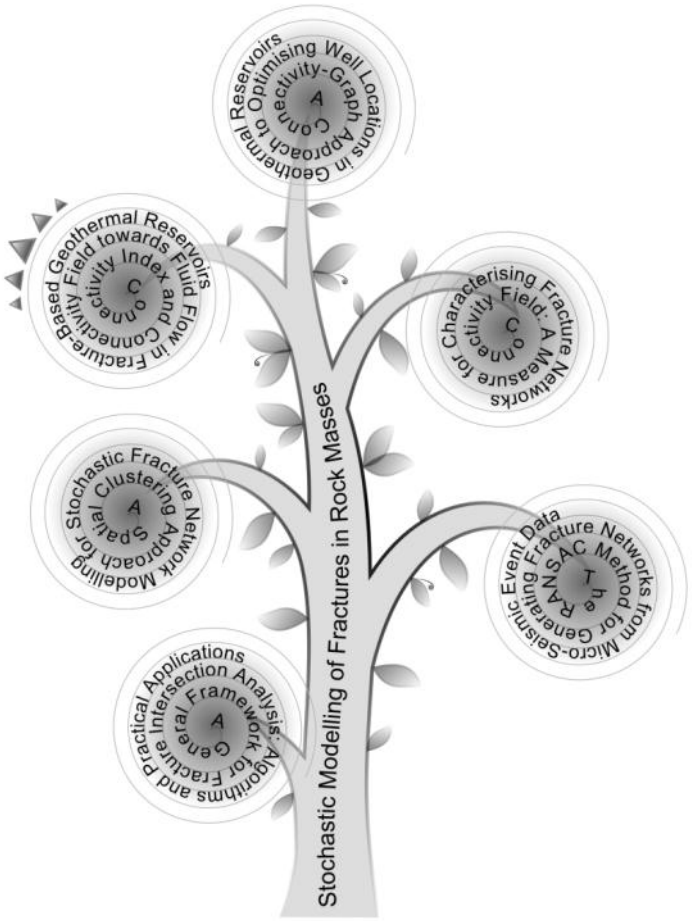
7.6 Concluding remarks

We propose the connectivity field (CF) and its extensions, generalised CF (GCF), probabilistic CF (PCF) and average CF ($\overline{\mathbf{CF}}$), as a means for characterising the connectivity of a fracture network both as deterministic measures for a given fracture network and as probabilistic measures for a stochastic fracture model. We have also given some potential applications of these measures for determining spatial fracture connectivity in a fracture network. CF values are calculated using the sampling locations in the study region (e.g., grid for systematic sampling) and the fracture clusters (generating hyper-clusters) in the network. As the connectivity information in CF is based on fracture intersections, the local and global relationships between fractures in the fracture network are preserved regardless of the size of the support being used for sampling. This is a distinguishing attribute of CF compared with lattice-based connectivity assessments. Higher CF values for a cell indicate a better connection of the cell to larger hyper-clusters of fractures in the reservoir. Such quantitative information can be useful in the design of drilling programmes to maximise performance for petroleum and geothermal reservoirs and aquifers. The CF can also be used to identify the main flow pathways through the fracture network. In addition, the CF can determine regions with minimum chance of fracturing based on a given fracture network, which can be useful for designing underground repositories. Finally, random sampling schemes (non-systematic sampling) for locating the supports can also be used in the evaluation of CF, which we will cover in future work.

Acknowledgements The work described here was funded by Australian Research Council Discovery Project grant DP110104766. We also thank Professor Rafael Jimenez and two other anonymous reviewers for their constructive comments.

8

Connectivity Index and Connectivity Field towards Fluid Flow in Fracture-Based Geothermal Reservoirs



This chapter is a conference paper of a practical nature dealing with the application of the connectivity index and connectivity field to geothermal reservoirs as a means of assessing potential fluid flow. Here CI and CF are used in fluid flow investigations. This chapter incorporates the fracture hydraulic properties (persistence/aperture) and utilises the finite difference method, FDM. The CF is demonstrated to be useful for characterising preferential flow pathways. Ranked flow pathways are possible using CF measures and an extension of CF, the probabilistic connectivity field (PCF). Major contributions of the paper include classification of flow pathways and improved understanding of flow – with benefits in EGS stimulation programs.

Statement of Authorship

Title of Paper	Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Fadakar-A Y, Xu C, Dowd P.A (2013) Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs. in: proceedings of 38 Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, p417-427.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	proposing and development of the concepts, conducting simulations and data analysis, developiong algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to development of concepts. Co-supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

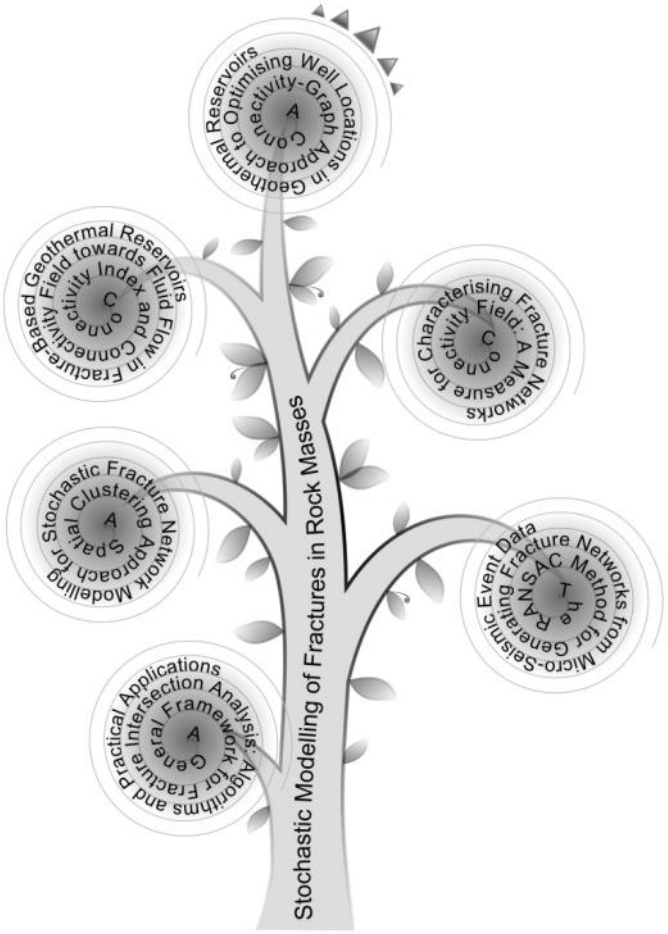
Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to development of concepts. Co-supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

Fadakar-A, Y., Xu, C. & Dowd, P.A. (2013) Connectivity index and connectivity field towards fluid flow in fracture-based geothermal reservoirs.
Presented at 38th Workshop on Geothermal Reservoir Engineering, Stanford, California, pp. 417-427

9

A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs



This chapter deals with the use of a connectivity graph approach to optimal well location in geothermal reservoirs and hence has mainly a practical focus. A distance-distribution function combined with the length and aperture of pathways is used (based on graph theory: *Dijkstra* shortest pathway algorithm) to optimise well location. A methodology using equivalent aperture and total length of pathway allows derivation of "weighted" shortest fluid pathways. An important contribution here is the ability to produce maps showing optimal well locations.

Statement of Authorship

Title of Paper	A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Fadakar-A Y, Dowd P.A, Xu C (2013) A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs. in: proceedings of Australian Geothermal Energy Conference AGEC2013, Brisbane, Australia, p111-115.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	Proposing and development of the concepts, conducting simulations and data analysis, developiong algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Co-supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

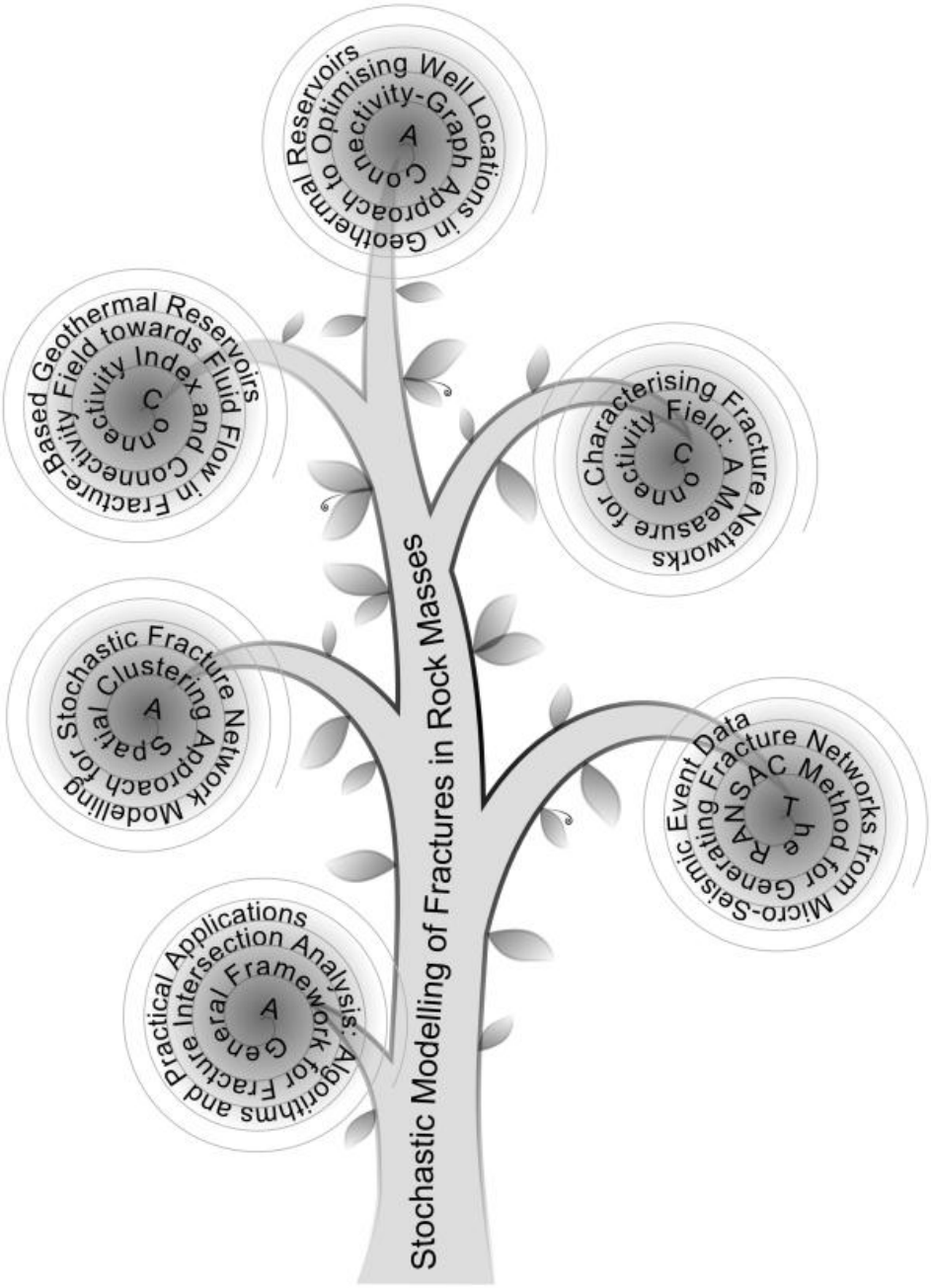
Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Co-supervising development of work.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

Fadakar-A, Y., Dowd, P.A. & Xu, C. (2013) A connectivity-graph approach to optimising well locations in geothermal reservoirs.
Presented at: Australian Geothermal Energy Conference, Brisbane, Australia, pp. 111-115

10

Concluding remarks


Stochastic Modelling of Fractures in Rock Masses
Optimising Well Locations in Geothermal Reservoirs: A Connectivity-Graph Approach
Fracture-Based Geothermal Reservoirs: Connectivity Index and Connectivity Field towards Fluid Flow
Characterising Fracture Networks for Connectivity Field: A Measure
Stochastic Fracture Network Modelling: Spatial Clustering Approach
Micro-Seismic Event Data Generating Fracture Networks from RANSAC Method
Practical Applications and Algorithms: Intersection Analysis for Fracture General Framework

10.1 Conclusions

Fractured rock is an important subject of interest in many applications including geothermal reservoirs, water aquifers, petroleum (oil and gas) production, and stability of underground works (e.g., tunnels). Fracturing process is also an important study in other areas of science and engineering such as geology (geosciences), material and mechanical engineering and manufacturing. As a particular application, modelling and simulation of fractures and their topological arrangement in the space (as fracture networks) is a key component of the assessment of hot dry rock (HDR) geothermal resources and of the design and creation of enhanced geothermal systems (EGS). The production of geothermal energy from an EGS heavily depends on fluid pathways through the HDR and thus fractures, fracture network and the interconnectivity of fractures in the network are essential factors (see Chap. 1 Introduction).

Rock mass comprises of the *matrix* and *fractures*. Fractures (as in its generalised definition) appear in wide ranges of scales and types, thus an ideally intact rock (i.e., with no fracture) does not exist in the real world. This identifies a need for better understanding of fractures and their characteristics such as interrelationships (fracture to fracture relationships) and patterns (spatial arrangement in the space). Generally, sampling of fractures (e.g., measuring fracture geometrical dimensions, its surface roughness, permeability and so on) is very costly if not possible in practice. Particularly, for EGS systems (kilometres beneath the ground) only indirect measurements e.g., geophysical investigations are possible. Due to these difficulties modelling techniques (including formulation process) are helpful for achieving this kind of understandings. A fracture in real situations can be filled by natural fillers (cements) such as products of crystallisation due to flowing fluid (e.g., in hydrothermal systems) or even by smaller size (compared to the aperture) rock particles (Figure 10.1). If a fracture is still void (or noticeably permeable otherwise) it has a determining role in directing the fluid flow within the rock mass due to the factors including gradient of permeability, inhomogeneity of in situ pressures and governing stress regimes (Sect. 00.00). Concisely, fracture network modelling is the study of fractures as a

system for evaluating its internal interactions between fractures and the resulting effect on the subject of study in local and global scales. Important example subjects are the fluid flow through fractures and the stability of rock blocks.

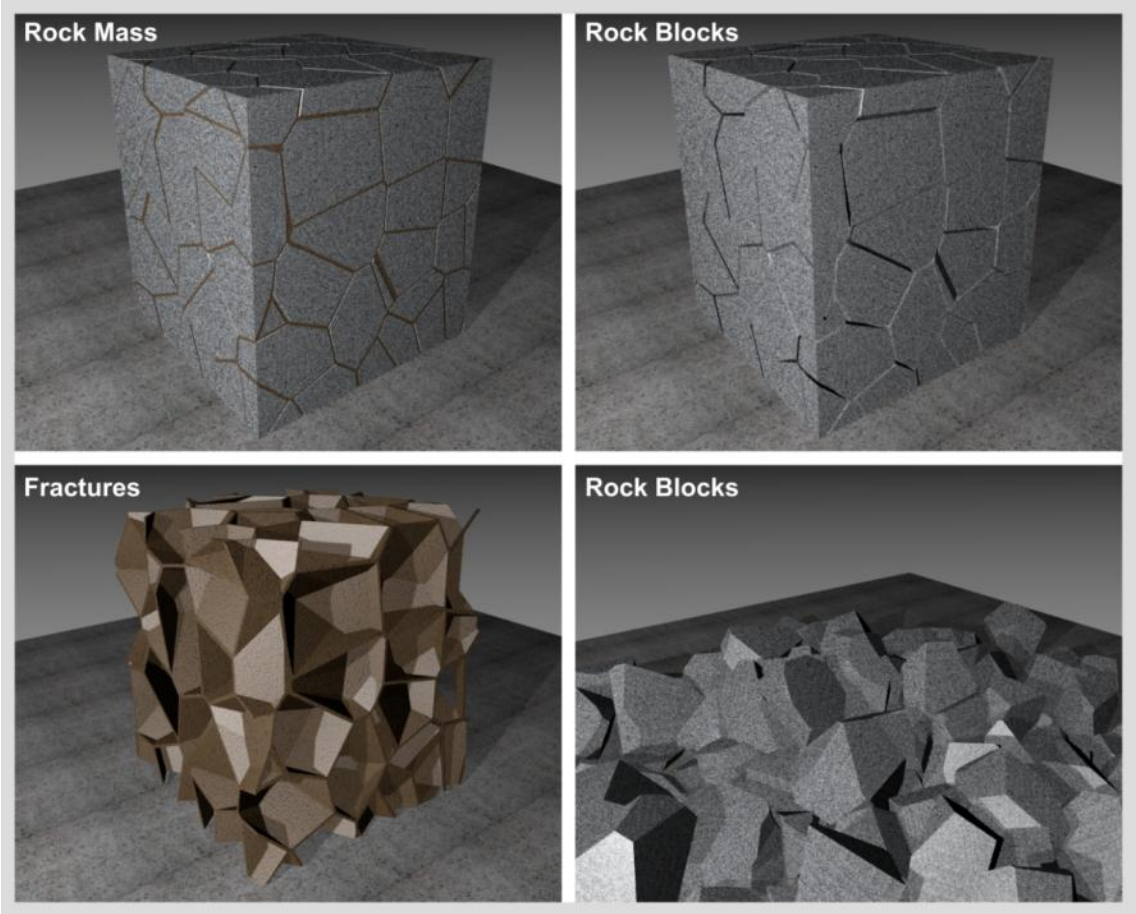


Figure 10.1: Rock mass: the matrix (blocks) and fractures. Modelling of fluid flow through fractures and the stability of rock blocks are two important applications of fracture network modelling.

In Chap. 2 (Fracture Network Modelling) fundamental terminology and definitions of the concept of fracture network modelling were presented followed by algorithms and discussions on the methodologies (including geometrical modelling) for simulating realistic fracture networks. It was shown that the developed DFN methods utilising stochastic concepts can mimic the complexity of geometry and topology of fractures and fracture networks observed in real world. Conditioning to the existing sampling data was also introduced. Some techniques for advanced modelling were proposed including the concept of growing fractures (Sect. 2.1.15) in which fractures are initialised as such either with negligible length or with a

desired length distribution. Then each fracture starts growing until intersection with another fracture or the boundary of study area occurs. This method can be linked somehow to the study of mechanics of fracture growth in materials. Furthermore, the proposed method has a potential for incorporating multiple growth rate depending on the initial fracture length or any other additional affecting factors, hence promises further applications. The growing method is capable to be conditioned to any existing fractures (e.g., faults) could result in desired fracture networks.

In Chap. 3 (A General Framework for Fracture Intersection Analysis: Algorithms and Practical Applications) further discussions were made on the methodologies to generate fractures and fracture networks satisfying the complexity of real fractures and fracture networks. As demonstrated any real and complex fracture (e.g., curvy fractures) can be decomposed into some flat polygons. Each flat polygon then is treated as an individual fracture. Since in subdividing process (decomposition) the relationships between fractures are preserved, therefore for any newly generated fracture the global topological information of the fracture network remains unchanged. Thus this method demonstrates capability to deal with any amount of complexity of real fractures in an efficient, robust and scalable manner. In this chapter, a robust framework to generate polygons in three-dimensional space which satisfies DFN requirements was proposed. It is worth noting that for computational performance all generated polygons are convex. For the reason that any convex polygon can be presented perfectly (with no loss of resolution, for example) by triangles (simplexes) it has been a standard in computational geometry codes for long time and is of interest in DFN codes as well. In DFN simulation, basically, fractures are modelled by convex flat polygons. Translating all generated fractures to their positions (determined by stochastic process e.g., Poisson point process) in the space; and then applying desired orientations (simulated based on directional statistical methods such as use of Fisher distributions), results in standard fracture networks in three-dimensions.

The above procedure can be further developed by means of Monte Carlo simulation method in order to characterise and determine the uncertainty associated with any individual parameter of the chosen models. The connectivity

between fractures in a fracture network determines the topology of the network. In order to analysis interconnection between fractures, in this chapter couple of efficient algorithms were discussed by which one can determine intersections between fractures, evaluate the resulting intersection points or lines and apply related statistical and spatial analysis. In addition, three novel analyses were introduced including a) intersection density as an effective and realistic representation measure of the connectivity (~conductivity) within a fracture network, b) distribution of the length of fracture intersection lines in a fracture network which is shown to be exponential, and c) the effect of length of fractures on the percolation state of two-dimensional fracture networks. It was shown that the number of percolation clusters has a non-linear relationship with the range of variations in the fracture length. In other words, fracture size and its unevenness are both important variables in percolation analysis and connectivity index evaluation, which in turn are important measures for the quantification of fluid flow characteristics of fracture networks.

The topic in Chap. 4 (The RANSAC Method for Generating Fracture Networks from Micro-Seismic Event Data) was basically on advancement in conditional simulation of three-dimensional fractures based on micro-seismic events measured during stimulation process in deep geothermal reservoirs. EGS are created from geothermal resources, usually located several kilometres below the surface of Earth, by establishing a network of connected fractures through which fluid can flow. The depth of the reservoir makes it impossible to conduct direct measurements of fractures and hence data are usually collected from indirect measurements such as geophysical surveys. An important source of indirect data is the seismic event point cloud generated by the fracture stimulation process. Locations of these points are estimated from recorded micro-seismic signals generated by fracture initiation, propagation or slip. Two methods for reconstructing realistic fracture trace lines and planes given the point cloud of seismic events data were described: Enhanced Brute-Force Search (EBFS) and RANSAC. The methods have been tested on a synthetic data set and on the Habanero dataset of Geodynamics' geothermal project in the Cooper Basin of South Australia. The results showed that the RANSAC method is an efficient and

suitable method for the conditional simulation of fracture networks. In this chapter, the RANSAC method was adapted for application to fracture network modeling from point data, in this case from micro-seismic events. It was also proposed an enhanced version of BFS by adding post-processing algorithms in which adaptations have been made to achieve better performance. EBFS has been presented as background for the RANSAC method and as a performance benchmark for it. Whilst EBFS provides an intuitive and straightforward approach to fitting fractures to point cloud data, its use is restricted to relatively small data sets; as the size of the point cloud increases the method rapidly becomes impractical due to the increase in computation, memory and time. RANSAC is a robust fitting method in general and, in particular, has been shown to be an efficient method for fitting fractures to point data, using reasonable computing time and memory. RANSAC performance was investigated for the effect of the choice of model parameters such as the distance tolerance for associating points with a candidate fracture and the number of trials per stage. A set of new assessment approaches with applications to quantify the resulting simulated fracture network in terms of the new similarity measure and the efficiency was also proposed. Case studies using simulated point cloud data in two- and three-dimensions and a real data set comprising seismic events resulting from a fracture stimulation process conducted in the Habanero geothermal energy system in 2003 were presented. In both types of point cloud the resulting fractures are very well matched either with the embedded fractures or with the interpretations in technical reports.

Discussions in Chap. 5 (A Spatial Clustering Approach for Stochastic Fracture Network Modelling) show that an important step in fracture network modelling is to estimate the number of fractures and the properties of individual fractures such as their size and orientation. In practice, due to the lack of data and the complexity of the problem, significant uncertainties associated with fracture network modelling exist. The objective in this chapter was to determine a fracture model from the conditioning data (micro-seismic events) by minimizing the sum of the distances of data points from the fitted fracture model. The model comprises a comprehensive fracture modification scheme based on simulated annealing and

also introduces a goodness-of-fit measure for the fitted fracture model. The measure utilises a fracture similarity and a clustering technique to find a locally optimal solution for fracture parameters. The method was applied to a simulated dataset and to a real 3D case study of the Habanero reservoir in the Cooper Basin, Australia. The results from the simulated dataset were satisfactory in terms of the objective function value and statistics including the histogram of fracture length and the rose diagram of fracture orientations (for 2D). The proposed method is general in the sense that it can be easily extended to 3D applications. The use of proposed DD-transform increases the accuracy as well as the efficiency of the modelling by introducing a locally optimal solution for fracture parameters. The results from the Habanero reservoir field data were also satisfactory in terms of the number of iterations, the number of fractures and point association (for 3D).

In the previous chapters the focus was to model fracture networks satisfying the complexity of real fractures and mimicking real fracture networks patterns. A big step further is to characterise fracture networks. The results of characterisation may serve comparison (matching test) between fracture networks or their responses (e.g., fluid flow), for example. Response matching evaluation helps to calibrate the modelling and simulation parameters in order to achieve desired and reliable realisations. Characterisation may depict hidden relationships between geometrical properties of the fractures and fracture networks and the response of the network against phenomena of study, e.g., fluid flow and stability of blocks. In Chap. 6 (Characterisation of Fracture Networks) in addition to investigations made on the existing characterising tools including density/intensity measures (FDC, X_f, intensity group), spatial measures (LEC, distance map, buffer effect, convex-hull), a couple of new proposals and developments were presented. Fracture density, a new proposal, evaluates accurately the density of fractures in a network. In two-dimensional fracture networks this measure is estimated by means of cell sampling on a grid covering the entire study area. The fracture density for any cell in the grid is calculated as the sum of the number of fractures wholly contained within the cell and the number that intersect the boundary of the cell. As a result, a matrix of fracture density (Fn) is produced. The resulting matrix can be easily mapped. It can readily therefore be used for conditioning fracture networks. These

measures can also be extended to three-dimensional fracture networks for which the sampling cell would be a cube. Inner-convex-hull applied to a point pattern determines the areas within the study area with the largest convex polygon that does not contain a point. This measure precisely provides the largest intact areas in the rock mass. Proposals also include new densities measures for backbone, block centroid, intersection, extended intersection, and fracture normal intersection. The relationships between principal stresses and the associated patterns in fracture networks in two- and three-dimensions were also investigated. Comparison of density measures was also conducted. Furthermore, another new measure called "Popularity Index" was introduced. Popularity index is calculated based on the number of intersections for each fracture in the network. Thus a larger number of intersecting fractures yields a higher popularity measure for fractures.

Connectivity Field as proposed in Chap. 7 (Connectivity Field: A Measure for Characterising Fracture Networks) helps to evaluate interconnectivity of fractures in fracture network in a unique way that the local and global connectivity information are preserved. Analysis of the connectivity of a fracture network is an important component of the design, assessment and development of fracture-based reservoirs in geothermal, petroleum and groundwater resource applications. It is a useful means of characterising the flow pathways and the mechanical behaviours of reservoirs. An appropriate practical measure is required for connectivity characterisation because of the extreme complexity of fracture networks. In this chapter, the connectivity field (CF), as a useful measure to evaluate the spatial connectivity characteristics of fractures in a fracture network was proposed. The CF can be applied on both a particular realisation of a fracture network model (for deterministic evaluation) and on stochastic fracture network models using stochastic modelling and Monte Carlo simulations (for probabilistic evaluation with uncertainties). Two extensions are also proposed: the generalised CF, a measure that is independent of support size, and the probabilistic CF. Potential applications of the CF and its extensions are in determining the optimal location of an injection or production well so as to maximise reservoir performance; and in determining potential flow pathways in fracture networks. The average CF map shows strong

correlations with the X_f and P21 measures. The relationships between the CF measures, the fracture intersection density and the fracture network connectivity index are also investigated.

The connectivity field (CF) and its extensions, generalised CF (GCF), probabilistic CF (PCF) and average CF ($\overline{\mathbf{CF}}$), as a means for characterising the connectivity of a fracture network both as deterministic measures for a given fracture network and as probabilistic measures for a stochastic fracture model were proposed. Some potential applications of these measures for determining spatial fracture connectivity in a fracture network were given. CF values are calculated using the sampling locations in the study region (e.g., grid for systematic sampling) and the fracture clusters (generating hyper-clusters) in the network. As the connectivity information in CF is based on fracture intersections, the local and global relationships between fractures in the fracture network are preserved regardless of the size of the support being used for sampling. This is a distinguishing attribute of CF compared with lattice-based connectivity assessments. Higher CF values for a cell indicate a better connection of the cell to larger hyper-clusters of fractures in the reservoir. Such quantitative information can be useful in the design of drilling programmes to maximise performance for petroleum and geothermal reservoirs and aquifers. The CF can also be used to identify the main flow pathways through the fracture network. In addition, the CF can determine regions with minimum chance of fracturing based on a given fracture network, which can be useful for designing underground repositories.

In Chap. 8 (Connectivity Index and Connectivity Field towards Fluid Flow in Fracture-Based Geothermal Reservoirs) the aim was to extend the use of developed measures (CI and CF) to real world applications such as fluid flow modelling. Connectivity measures including CI and CF are useful for determining preferential flow directions and flow pathways through fracture networks. However, the current implementation of these measures does not consider the hydraulic properties of the fracture network, which is the issue addressed in this chapter. Darcy's law can be incorporated into the evaluation of these measures using the persistence and aperture properties of fractures in the fracture network. This incorporation helps determine more reliable and accurate flow pathways in the

fracture network for three forms of aperture distributions: (*i*) constant aperture for each fracture cluster (pathway), (*ii*) variable aperture for each cluster and (*iii*) variable aperture for each fracture. A new concept for the classification of pathways based on the reliability of their assessment was introduced, which enhances the understanding of the flow behaviour of fracture-based reservoirs as a result of fracture network expansion processes such as hydraulic stimulation.

Briefly, in this chapter a method to incorporate the size and aperture of fractures in the connectivity field and connectivity index was introduced. It was shown that incorporating these properties makes it possible to link much more realistically the connectivity of fracture networks to fluid flow (using a method analogous to Darcy's law), which in turn significantly increases the reliability of the connectivity analysis. The application of the proposed measures in directional flow analysis with the benefit of increased accuracy and reliability were discussed. The rose diagrams resulting from CI and WCI were compared to results from the finite difference method. It was also shown that the incorporation of the length and aperture of fractures improves significantly the consistency between the results from the connectivity measure and the finite difference solutions. The application of CF to the direct determination of flow pathways in fracture networks and to providing a ranking system for pathways that can be used to predict and assess the extent of fracture networks as the result of a multi-stage hydraulic fracture stimulation process was also demonstrated.

Yet another new application for connectivity measures was discussed in Chap. 9 (A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs) where the problem of optimal location of injection and production wells in fractured-based geothermal reservoirs was addressed. The optimisation was based on a distance distribution function, and the length and aperture of pathways between the two wells. The initial locations of the two candidate wells are chosen at random and the fracture pathways between the wells are determined using graph theory concepts. The weighted shortest pathway incorporates the equivalent aperture and total length of pathway elements (i.e., linked fractures). The method is efficient and effective for generating final optimal well locations (as coordinates) and also provides a map of optimality for any given fracture network. The sampling

scheme used can incorporate any constraint including technical, topographical and or design. Furthermore, stochastic modelling of fracture networks can be used to extend the use of the proposed method to deal with the uncertainty involved in estimated or simulated fracture networks.

In summary, the proposed method is a simple but effective and realistic solution for determining optimal well locations for fracture-based reservoirs such as enhanced geothermal systems. It is based on generating random point pairs separated by a distance drawn from a distance distribution function, for example $\mathbf{N}(\mu, \sigma)$. The parameters of the distance distribution are defined on the basis of the technical, topographical and design requirements. The proposed method uses regional and local backbone extraction for the fracture network. The pathways are determined by means of graph theory algorithms such as the *dijkstra* method. Each element of the local backbone is associated with a weight corresponding to its length and aperture, which in turn result in an equivalent resistivity (analogous to inappropriateness of the pathway). Note that while the length of the element is calculated locally (necessarily due to intersections and the application of backbone procedure) its aperture value is taken from the original fracture network. A number of trials are conducted using Monte Carlo line sampling for each local backbone and the shortest weighted pathways are extracted. Finally, the minimum total pathway weight suggests the optimal well locations. Our primary goal in the present work was to examine, evaluate and demonstrate the usefulness of the proposed method. The method is readily adaptable to transmissivity and other problems. The conceptually simple formulation of the method offers promise for further development and applications.

10.2 Summary of contributions

Contributions of this thesis can be summarised as follows.

10.2.1 Journal Papers

1. The RANSAC method for generating fracture networks from micro-seismic event data (*J Mathematical Geosciences,* published)

2. Connectivity Field: A measure for characterising fracture networks (*J Mathematical Geosciences,* published)

3. A Spatial Clustering Approach for Stochastic Fracture Network Modelling (*J Rock Mechanics and Rock Engineering,* published)

4. Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs (*J Mathematical Geosciences,* under review)

10.2.2 Conference Papers and Presentations

5. A general framework for fracture intersection analysis: algorithms and practical applications (*AGEC2011,* published)

6. Application of Connectivity Measures in Enhanced Geothermal Systems (*AGEC2012,* published)

7. Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs (*SGW2013,* published)

8. A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs (*AGEC2013,* published)

10.2.3 Talk

9. Applications of connectivity measure in fracture-based reservoirs for: fluid flow modeling, optimal drilling locations design and reservoir characterization (*University of British Columbia, Canada, UBC2013*)

10.2.4 Developed and new terms, concepts, models, algorithms, frameworks and methods

The following list provides section reference to the contributions (from this research) which are either novel (original) or developed. It is worth noting that developments also provide some extent of novelty. Ultimately, the main purpose was and is to enrich and empower simulation (generating), characterisation and application (the three major stages) of fracture networks.

- Fracture growth concept (Sect. 2.1.15); Growing conditioned to existing fractures (Sect. 2.1.15); Additional processing stages for fracture network modelling (Sect. 2.2); Fracture extension and trimming (Sect. 2.4); Efficient algorithm to model n-edge convex polygons (Sect. 2.5); Conditioning fracture locations using Simulated Annealing (Sect. 2.6.1); Conditioning to existing fractures (Sect. 2.6.2); Application of Hough transform in fracture network modelling (Sect. 2.6.3); Application of RANSAC in fracture network modelling (Sect. 2.6.3)
- Framework to generate realistic fracture network by means of marked point processes (Sect. 3.2); Locations from Poisson point processes (Sect. 3.2.1); Fracture circumscribe rectangle side sizes from exponential distributions (Sect. 3.2.3); Intersection Analysis in 3D (Sect. 3.3); A full robust framework for fracture-fracture intersection analysis (Sect. 3.3); Intersection Density (Sect. 3.3.1 and 6.13.2); Lengths of Intersection Lines (Sect. 3.3.2); Effects of Fracture Length on Percolation State (Sect. 3.3.3)
- Enhanced Brute-Force Search (Sect. 4.2.1); Line Cluster Fitting (LCF, Sect. 4.2.1); RANSAC algorithm for fracture network modelling (Sect. 4.2.2); Performance of RANSAC for line and plane fitting (Sect. 4.2.2); Fracture Extents (Sect. 4.2.3); A flexible procedure to generate a two-dimensional point cloud (Sect. 4.3.1); Algorithm for fracture networks similarity assessment (Sect. 4.3.1)
- A Spatial Clustering Technique (Sect. 5.4); Algorithm of DD-transform (Sect. 5.4)
- Fracture centroid density (Sect. 6.1); Fracture density (Sect. 6.4); Largest empty circle (Sect. 6.5); Largest empty convex-hull (inner-convex-hull, Sect. 6.6); Distance map (Sect. 6.7); Buffer effect (Sect. 6.8); Convex-hull (Sect. 6.9); Block area (Sect. 6.10); Backbone density (Sect. 6.11); Block centroid density (Sect. 6.12); Intersection analysis (Sect. 6.13); Fracture clusters and pipe modelling (Sect. 6.13.1); Extended intersection density (Sect. 6.13.3); Fracture normal intersection density (Sect. 6.13.5); Effects of stress field (Sect. 6.14); Popularity index (Sect. 6.16)
- Connectivity Field: A measure for characterising fracture networks (Chap. 7 and Sect. 7.2); Fracture/support-based connectivity (Sect. 7.1); Fully

isolated fracture networks (Sect. 7.1); The Generalised Connectivity Field (Sect. 7.2.1); The Probabilistic Connectivity Field (Sect. 7.2.2); The relationships between the CF, GCF, DFC, ID, X f and P21 (Sect. 7.3); Average CF (Sect. 7.3); CF and Flow Pathways (Sect. 7.4.1); CF and Percolation State (Sect. 7.4.2); Using CF to determine well locations and to design underground repositories (Sect. 7.4.3); CF applied to a real three-dimensional fracture network (Sect. 7.5)

- Connectivity Index and Connectivity Field towards Fluid Flow in Fracture-Based Geothermal Reservoirs (Chap. 8); Model to extract directional flow information (Sect. 8.2.2); The framework for a finite difference method used to model the flow through fracture networks in two dimensions (Sect. 8.2.2); Incorporating Length and Aperture in CI: WCI (Sect. 8.3.2); A weighting factor based on the length and aperture of fractures forming the pathways between two supports (Sect. 8.3.2); Preferential Flow Pathways using CF (Sect. 8.4.1); Histograms of orientation errors for CI and WCI (Sect. 8.4.1); CI Field (Sect. 8.4.1); Relationship between PCF and CIF (Sect. 8.4.1)
- A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs (Chap. 9); Generic form of equivalent aperture (Sect. 9.2.1)
- Matlab package for fracture network modelling (Appendix 2)

The above list of contributions can easily be matched with the three main stages in this research: generation, characterisation and application stages. For example, fracture growth concept, extension and trimming algorithms, Hough transform and RANSAC methods are examples of important contributions in the stage of generation (simulation) of fracture networks. Fracture networks similarity measure, intersection analysis including intersection density, fracture density, popularity index and other proposed measures together with Connectivity Field group are main contributions to the field of fracture network characterisation. Weighted CF and CI measures in which length and aperture of fractures are incorporated and application of graph theory concepts provide useful tools for extending the proposed theoretical concepts into real world applications; and are

together with their example applications in EGS main contributions. Classification of flow pathways; improved understanding of flow – with benefits in EGS stimulation programs; and the ability to produce maps showing optimal well locations are other contributions. As another noticeable contribution the Matlab code package provided in Appendix 2 is useful to future DFN engineering researchers.

10.3 Ideas for future research

The following ideas are of interest for future developments:

- Developing an extension of CF in which the edge effect due to grid sampling is minimised. The proposal is to centre a grid covering study area on a randomly positioned target cell (implementation of random sampling scheme) and then to apply standard CF computation for the cell. In this way due to unrestricted (non-grid) movement of the target cell in the study region the resulting CF will have no cell edge effect. The method shall be called Random Connectivity Field (RCF). A simple demonstration of its implementation is shown in Figure 10.2. With no additional work or complexity this idea can easily be extended into three-dimensions where the position of the target cell will be a tuple of three random coordinates $p_{tcell} = (x, y, z)$. Note that fracture clustering information as an instant result of intersection analysis on the fracture network plays an important role in optimising the performance of evaluation of intersections between the newly generated grid and the network at each stage. Other optimisations for improving computing performance are also to be investigated. Random sampling scheme may open further opportunities to study the effect of pattern of sampling points on the resulting CF maps and shall suggest new extensions.

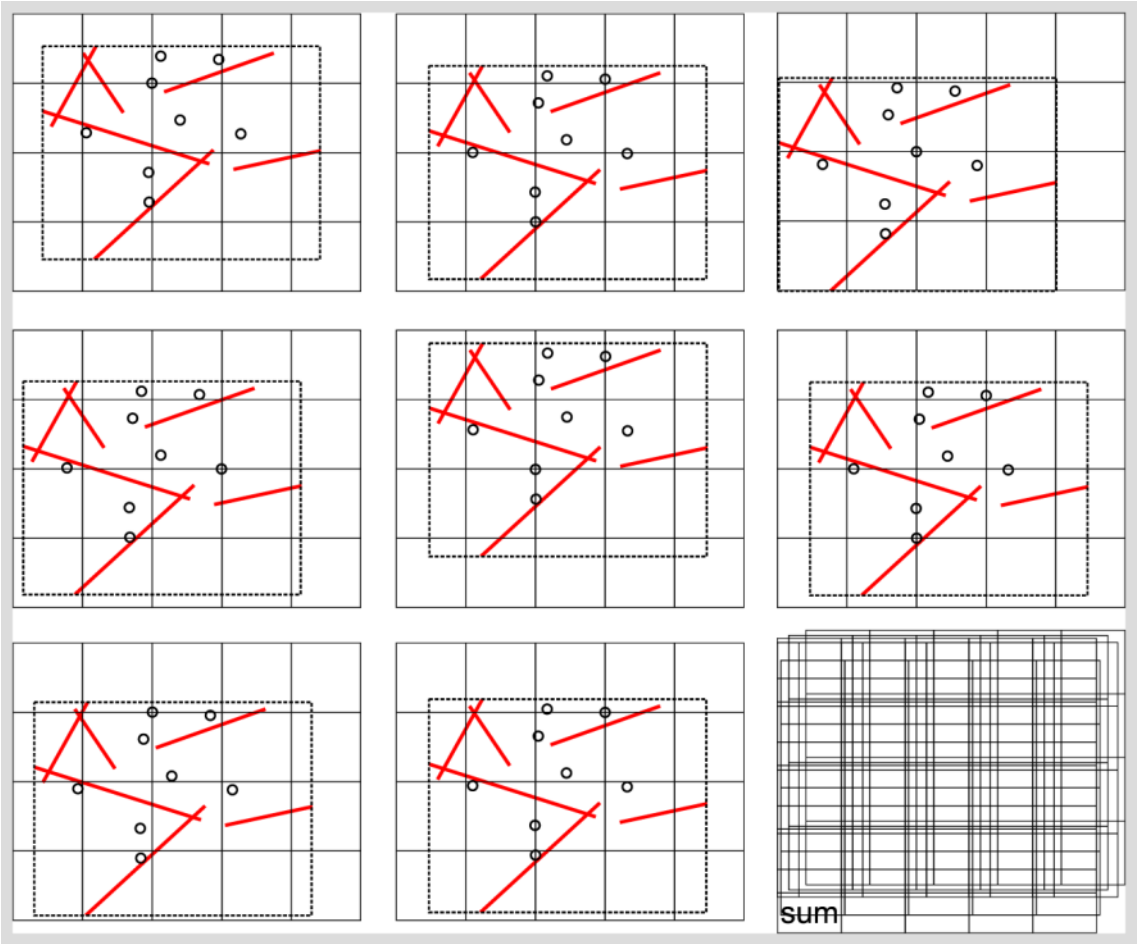


Figure 10.2: RCF. The grid is centred on each point as shown. CF is calculated for each move. RCF is the sum of all resulting CF maps.

- Investigations on the association of CF with Geological Strength Index (GSI), a matter of interest in rock mechanics according to feedback received from the presentations in SGW2013 and UBC2013.

- Further development of mathematical relationships between CF, CI, and other fracture network characterisation measures are to be done. The relationship between PCF and CIF as extracted in Fadakar-A et al. (2013b, SGW2013) can be a starting point.

- Incorporation of length and aperture in the evaluation of connectivity measures as proposed in Fadakar-A et al. (2013b-2013c) can be further developed. Concepts including the number of pathways between the two connected supports, the complexity of the pathways, a deeper

incorporation of the mechanics of fluid flow (effects of gravitational force, capillarity etc.) are the areas for further developments in the use of weighting system for connectivity assessment.

Appendices

1

Application of Connectivity Measures in Enhanced Geothermal Systems

Statement of Authorship

Title of Paper	Application of Connectivity Measures in Enhanced Geothermal Systems.
Publication Status	◉ Published, ○ Accepted for Publication, ○ Submitted for Publication, ○ Publication style
Publication Details	Fadakar-A Y, Dowd P.A, Xu C (2012) Application of Connectivity Measures in Enhanced Geothermal Systems. in: proceedings of Australian Geothermal Energy Conference AGEC2012, Sydney, Australia, p62-66.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to the publication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Name of Principal Author (Candidate)	Younes Fadakar Alghalandis		
Contribution to the Paper	Proposing and development of one of the concepts (connectivity field) used in the paper, conducting simulations and data analysis, developing algorithms, interpretation of the results, producing and preparing figures, writing the manuscript, preparing required publication format, submitting, acting as corresponding author.		
Signature		Date	28 02 2014

Name of Co-Author	Peter Alan Dowd		
Contribution to the Paper	Contributions to development of concepts and major contributions to previously developed concepts (connectivity index) used in the paper. Supervising development of work, manuscript evaluation and editing.		
Signature		Date	28/2/2014

Name of Co-Author	Chaoshui Xu		
Contribution to the Paper	Contributions to development of concepts and major contributions to previously developed concepts (connectivity index) used in the paper. Assistance in editing and evaluation of the manuscript.		
Signature		Date	28/2/2014

Name of Co-Author	N/A		
Contribution to the Paper	N/A		
Signature		Date	

Fadakar-A, Y., Dowd, P.A. & Xu, C. (2012) Application of connectivity measures in enhanced geothermal systems.
Presented at: Australian Geothermal Energy Conference, Sydney, Australia, pp. 62-66

2

Matlab Programs: Alghalandis Fracture Network Modelling (AFNM) Package

2.1 Introduction

Alghalandis Fracture Network Modelling (**AFNM**) is a package of computer codes in Matlab language syntax which consists of a) functions to generate fracture networks in two and three dimensions based on stochastic modelling principals (e.g., discrete fracture network modelling framework); b) functions to characterise synthesised or imported two- and three-dimensional fracture networks including intersection analysis, density measures, connectivity indices, clustering and many others; c) functions for highly simplification of visualisation of two- and three-dimensional fracture networks; and d) functions to generically utilise the above stages and to extend their use for practical applications, to provide stable framework for further developments, and tools to save the resulting maps, tables and information in appropriate formats readable by many common standard software applications. The intension of this endeavour was to unify the computation stages i.e., framework and even more the core functions such to get the highest productivity. With the hope readers will find this package handy and useful for evaluation of the concepts proposed in this thesis, developing their own ideas and experiencing fracture network modelling concepts.

A couple of external Matlab single codes or packages are linked here without inclusion of their code including "*geom2d*[10]" and "*geom3d*[11]" (edition 2011, by David Legland), "*circstat*[12]" (edition 2011, by Philipp Berens), "*kde2d.m*[13]" (edition 2009, by Zdravko Botev), "*smoothn.m*[14]" (edition 2010, by Damien Garcia), "*vol3d.m*[15]" (edition 2009, by Oliver Woodford), and "*dict.m*[16]" (edition 2008, by Doug Harriman).

[10] http://www.mathworks.com.au/matlabcentral/fileexchange/7844-geom2d
[11] http://www.mathworks.com.au/matlabcentral/fileexchange/24484-geom3d
[12] http://www.mathworks.com.au/matlabcentral/fileexchange/10676-circular-statistics-toolbox-directional-statistics
[13] http://www.mathworks.com.au/matlabcentral/fileexchange/17204-kernel-density-estimation
[14] http://www.mathworks.com.au/matlabcentral/fileexchange/25634-easy-n-fast-smoothing-for-1-d-to-n-d-data
[15] http://www.mathworks.com.au/matlabcentral/fileexchange/22940-vol3d-v2
[16] http://www.mathworks.com.au/matlabcentral/fileexchange/19647-dict

2.2 License

The provided computer codes in this chapter are copyrighted. They are made publically available for use under the following license and conditions.

2.3 List of functions

Name	Description	Name	Description
`Angles2D`	*angles of 2D lines*	`Centers2D`	*centres of 2D lines*
`Lengths2D`	*lengths of 2D lines*	`GenFNM2D`	*2D fracture network*
`ClipLines2D`	*clips 2D lines*	`LinesXLines2D`	*two line sets intersections*
`LinesX2D`	*intersection analysis on 2D lines*	`LinesToClusters2D`	*clusters of lines*
`Density2D`	*true density of 2D lines*	`Histogram2D`	*2D histogram (density)*
`RandLinesInPoly2D`	*random line sampling*	`Sup2D`	*creates 2D support*
`SupCSup2D`	*two 2D supports' connectivity*	`SupXLines2D`	*intersections between a support and 2D lines*
`SupXNLines2D`	*sup intersects 2D lines*	`P21G`	*P21 gridded measure*
`ConnectivityIndex2D`	*CI*	`ConnectivityField2D`	*CF*

BreakLinesX2D	*break lines at their intersections*	Rotate2D	*rotates 2D points*
SortPoints2D	*topological sort of points*	Isolated2D	*checks if a 2D line isolated*
Backbone2D	*backbone of 2D lines*	IsolatedLines2D	*checks isolation for all 2D lines*
BackboneToNodesEdges2D	*backbone to graph*	Expand2D	*expands 2D matrix*
Resize2D	*resize 2D matrix*	DrawLines2D	*draws 2D lines*
LinesToXYnan2D	*convert lines to X and Y*	ExpandAxes2D	*expands 2D axes*
Titles2D	*title, labels, grid … for axes in 2D*	RandPoly3D	*random 3D polygons*
GenFNM3D	*3D facture network*	Sup3D	*creates 3D support*
ClipPolys3D	*clips 3D polygons*	PolysX3D	*intersection analysis on 3D polygons*
PolysXPolys3D	*polygons' intersections*	PolyXPoly3D	*intersection between two 3D polygons*
SupCSup3D	*two 3D supports' connectivity*	BBox3D	*bounding box of 3D points*
Expand3D	*expand 3D matrix*	Resize3D	*resizes 3D matrix*
SaveToFile3D	*save 3D result to file*	SavePolysToVTK3D	*saves 3D polygons as VTK file*
SetAxes3D	*sets 3D axes*	DrawPolys3D	*draws 3D polygons*
DrawSlices3D	*draws 3D slices*	VolRender3D	*draws volume render of 3D volumetric data*
Scale	*scales data*	ToStruct	*converts to `struct` format*
Clusters	*cluster analysis*	CheckClusters	*checks clusters for errors*
Labels	*labels*	Relabel	*relabels cluster labels*
Stack	*stacks cell data*	Group	*groups data based on common elements*
FarthestPoints	*two farthest points*	PDistIndices	*generates indices for `pdist` function*
Occurrence	*occurrence of points*	ConnectivityMatrix	*connectivity matrix (CM)*
FullCM	*full form of CM*	FNMToGraph	*converts fracture network to graph*
LoadColormap	*loads colormap*	SaveColormap	*saves current colormap*
SecondsToClock	*seconds to clock*	Colorise	*colourise given data*
ShowFNM	*shows fracture network*	Round	*rounds data*

2.4 Functions for Two Dimensional Cases

2.4.1 Angles2D

```
% Angles2D
% returns angles of 2D lines (fracture)
%
% Usage :
%    ags = Angles2D(lines)
%
% input : lines     (n,4)
% output: ags       (n) in radian
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function ags = Angles2D(lines)
ags = atan2(lines(:,4)-lines(:,2),lines(:,3)-lines(:,1));
```

Angles2D computes orientation angles of two-dimensional fractures (line segments) with correct sign for each quadrant. Lines are represented as a two-dimensional array of size $n \times 4$ here and in all other functions.

2.4.2 Centers2D

```
% Centers2D
% returns center points of 2D fracture lines
%
% Usage :
%    cts = Centers2D(lines)
%
% input : lines     (n,4)
% output: cts       (n,2)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function cts = Centers2D(lines)
cts = 0.5*[lines(:,3)+lines(:,1),lines(:,4)+lines(:,2)];
```

Centers2D computes centre of fracture lines. The centring points can be used to compute the **DFC (FCD)**, for example.

2.4.3 Lengths2D

```
% Lengths2D
% returns lengths of 2D fracture lines
%
% Usage :
%    lhs = Lengths2D(lines)
%
```

```
% input : lines     (n,4)
% output: lhs       (n)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function lhs = Lengths2D(lines)
if isempty(lines)
    lhs = 0;
else
    lhs = hypot(lines(:,3)-lines(:,1),lines(:,4)-lines(:,2));
end
```

Lengths2D calculates the Euclidean lengths between the two endpoints of lines. This function can be used for estimation of **P21** measure, for example.

2.4.4 GenFNM2D

```
% GenFNM2D
% generates 2D fracture network
%
% Usage :
%    [lines,olines] = GenFNM2D(n,theta,kappa,minl,maxl,rgn)
%
% input : n         number of fracture lines, default=150
%         theta     main orientation, default=0
%         kappa     Fisher dispersion factor, default=0: omnidirectional
%         minl      minimum length of fracture lines, default=0.05
%         maxl      maximum length of fracture lines, default=1
%         rgn       region of study, default=[0,1,0,1] i.e., unit square
% output: lines     fracture lines after clipping by rgn
%         olines    original lines
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [lines,olines] = GenFNM2D(n,theta,kappa,minl,maxl,rgn)
if nargin<6; rgn = [0,1,0,1]; end
if nargin<5; maxl = 1; end
if nargin<4; minl = 0.05; end
if nargin<3; kappa = 0; end
if nargin<2; theta = 0; end
if nargin<1; n = 150; end
pts = rand(n,2);                                %locations~ U(0,1)
ags = circ_vmrnd(theta,kappa,n);                %oreint.~von-Mises(theta=0,kappa=0)
lhs = Scale(exprnd(1,n,1),minl,maxl);           %lengths~ Exp(mu=1)
[dx,dy] = pol2cart(ags,0.5*lhs);
olines = [pts(:,1)-dx,pts(:,2)-dy,pts(:,1)+dx,pts(:,2)+dy];  %original
lines = ClipLines2D(olines,rgn);                %clipped by region of study
```

GenFNM2D synthesises two-dimensional fracture networks according to stochastic modelling principals in which for locations, orientations and lengths finite samples from desired random distribution functions are drawn. In this implementation for

locations a simple uniform distribution is chosen while for orientations von-Mises distribution is used. The lengths are generated based on an exponential distribution and are truncated into a specific range by scaling. In use, one may adapt the code easily to benefit from other distributions.

2.4.5 ClipLines2D

```
% ClipLines2D
% returns clipped 2D fracture lines by a given rectangle (box)
%
% Usage :
%    clines = ClipLines2D(lines,box)
%
% input : lines     (n,4)
%         box       (4), default=[0,1,0,1]
% output: clines    (n,4)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function clines = ClipLines2D(lines,box)
if nargin<2; box = [0,1,0,1]; end
[m,n] = size(lines);
clines = zeros(m,n);
for i = 1:m
    clines(i,:) = clipEdge(lines(i,:),box);
end
```

ClipLines2D provides handy tool to apply clipping to two-dimensional fracture lines by a given rectangle.

2.4.6 LinesXLines2D

```
% LinesXLines2D
% finds intersection indices and points between two sets of 2D fracture lines
%
% Usage:
%    [xpss,idss] = LinesXLines2D(lines1,lines2)
%
% input : lines1    (m,4)
%         lines2    (n,4)
% output: xtss      intersection points, (cell)
%         idss      intersecting lines' indices, (cell)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [xpss,idss] = LinesXLines2D(lines1,lines2)
m = size(lines1,1);
n = size(lines2,1);
idss = cell(m,1);
xpss = cell(m,1);
```

```
u = 0;
for i = 1:m                                          %for all lines in lines1
    idx = zeros(1,1);
    xps = zeros(1,2);
    found = false;
    k = 0;
    for j = 1:n                                      %for all lines in lines2
        xpt = intersectEdges(lines1(i,:),lines2(j,:));
        if ~isfinite(xpt(1)); continue; end
        k = k+1;
        idx(k) = j;
        xps(k,:) = xpt;
        found = true;
    end
    if found                                         %record if there was any
intersection
        u = u+1;
        idss(u) = {{i,idx}};
        xpss(u) = {xps};
    end
end
idss = idss(1:u);                                    %compaction
xpss = xpss(1:u);
```

LinesXLines2D is made available to apply intersection analysis between two sets of lines. The resulting intersection points and the identity number of intercepting lines are reported in the format of Matlab "cell" data type. The implementation is optimised to consume minimum memory requirement and also to deliver high performance.

2.4.7 LinesX2D

```
% LinesX2D
% finds intersection indices and points for a set of 2D fracture lines
%
% Usage :
%    [xts,ids,La] = LinesX2D(lines)
%
% input : lines     (n,4)
% output: xts       intersection points, (m,2)
%         ids       intersecting lines indices, (m,2)
%         La        cluster labels (n)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [xts,ids,La] = LinesX2D(lines)
n = size(lines,1);
m = n*(n-1)/2;                                  %max possible number of intersections
xts = zeros(m,2);
ids = zeros(m,2);
k = 0;
for i = 1:n-1                                        %apply optimum iteration
    for j = i+1:n
        xpt = intersectEdges(lines(i,:),lines(j,:));
```

```
        if ~isfinite(xpt(1)); continue; end
        k = k+1;
        xts(k,:) = xpt;                          %intersection points
        ids(k,:) = [i,j];                        %intersecting lines' indices
    end
end
xts = xts(1:k,:);                                %compaction
ids = ids(1:k,:);
La = Labels(Clusters(num2cell(ids,2)),n);        %fracture cluster labels
```

LinesX2D is to compute intersections between all lines in a set of fracture lines. This function also computes fracture clusters with appropriately assigned labels. It plays an important role in analysing connectivity measures, for example.

2.4.8 LinesToClusters2D

```
% LinesToClusters2D
% finds clusters of 2D fracture lines (inter-connections)
%
% Usage :
%    La = LinesToClusters2D(lines)
%
% input : lines     (n,4)
% output: La        (n) cluster labels
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function La = LinesToClusters2D(lines)
[~,~,La] = LinesX2D(lines);
```

LinesToClusters2D wraps **LinesX2D** for cases in which only label of clusters are required rather than complete intersection information.

2.4.9 Density2D

```
% Density2D
% computes true density of 2D fracture network
%
% Usage :
%    [DN,x,y] = Density2D(lines,gm,gn)
%
% input : lines     (n,4)
%         gm        grid dimension vertically
%         gn        grid dimension horizontally
% output: DN        (gm,gn)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [DN,x,y] = Density2D(lines,gm,gn)
w = 1/gn;                                        %sampling cell's width
h = 1/gm;                                        %sampling cell's height
```

```
DN = zeros(gm,gn);
for i = 1:gm
    for j = 1:gn
        sup = [(j-1)*w,j*w,(i-1)*h,i*h];
        DN(i,j) = SupXNLines2D(sup,lines);
    end
end
x = [w/2,1-w/2];                                    %for usage in imagesc(x,y,DN)
y = [h/2,1-h/2];
```

Density2D computes the true density of fracture lines in the region of study based on a grid ($gm \times gn$) sampling.

2.4.10 Histogram2D

```
% Histogram2D
% computes 2D histogram (~density) of points
%
% Usage :
%    out = Histogram2D(pts,nx,ny)
%
% input : pts       (n,2)
%         nx        grid dimension on axis X
%         ny        grid dimension on axis Y
% output: out       (ny,nx)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function out = Histogram2D(pts,nx,ny)
if nargin<3
   nx = 7;                                          %default grid size
   ny = 7;
end
k = [nx,ny];
bin = zeros(size(pts));
for i = 1:2                                         %iterate for X and Y coordinates
    minx = min(pts(:,i));
    egs = minx+((max(pts(:,i))-minx)/k(i))*(0:k(i));
    [~,t] = histc(pts(:,i),[-Inf,egs(2:end-1),Inf],1);
    bin(:,i) = min(t,k(i));
end
out = accumarray(bin(all(bin>0,2),:),1,k)';
```

Histogram2D provides a superbly fast evaluation of density of points based on the concept of histogram classification. For some cases this function is superior to **KDE** estimation if the local variation is more important than global smoothing.

2.4.11 RandLinesInPoly2D

```
% RandLinesInPoly2D
% generates random 2D sampling lines inside a 2D polygon
%
% Usage :
```

```
%    lines = RandLinesInPoly2D(n,h,dh,ang,da,poly)
%
% input : n         number of lines
%         h         length of lines
%         dh        length's tolerance
%         ang       angle
%         da        angle tolerance
%         poly      (k,2)
% output: lines     (n,4)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function lines = RandLinesInPoly2D(n,h,dh,ang,da,poly)
if nargin<6; poly = [[0,0];[1,0];[1,1];[0,1]]; end      %default: unit square
if nargin<5; da = 0; end
if nargin<4; ang = 0; end
if nargin<3; dh = 0; end
if nargin<2; h = 0.1; end
if nargin<1; n = 100; end
px = poly(:,1);
py = poly(:,2);
lines = zeros(n,4);
i = 0;
while i<n
    x1 = rand(1,1);
    y1 = rand(1,1);
    rr = (2*rand(1,2)-1);
    [dx,dy] = pol2cart(ang+rr(1)*da,h+rr(2)*dh);
    x2 = x1+dx;
    y2 = y1+dy;
    if inpolygon([x1,x2],[y1,y2],px,py)           %if line is inside the polygon
        i = i+1;
        lines(i,:) = [x1,y1,x2,y2];
    end
end
```

RandLinesInPoly2D is to generate n random lines inside of a given polygon. The length of lines is defined by a scalar h with variation dh. The orientation follows scalar ang with tolerance of da both in radian.

2.4.12 Sup2D

```
% Sup2D
% creates a support at point(x,y) with width(w) and height(h)
%
% Usage :
%    sup = Sup2D(x,y,w,h)
%
% input : x,y       coordinates of the center of support
%         w,h       width and height
% output: sup       (4,4), a rectangle
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
```

```
% Updated: Nov 2013
function sup = Sup2D(x,y,w,h)
sup = [[0,0,w,0];[w,0,w,h];[w,h,0,h];[0,h,0,0]];
sup(:,[1,3]) = sup(:,[1,3])+x-0.5*w;
sup(:,[2,4]) = sup(:,[2,4])+y-0.5*h;
```

Sup2D generates a two-dimensional support (here rectangle).

2.4.13 SupCSup2D

```
% SupCSup2D
% test for connectivity between two 2D supports via fracture network
%
% Usage :
%    C = SupCSup2D(lines,La,sup1,sup2)
%
% input : lines     (n,4), fracture network
%         La        (n) cluster labels for lines
%         sup1      support 1, box,e.g., [0,1,0,1]
%         sup2      support 2
% output: C         true/false
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function C = SupCSup2D(lines,La,sup1,sup2)
xC1 = SupXLines2D(sup1,lines,La);                  %cluster info of sup1 from fnm
xC2 = SupXLines2D(sup2,lines,La);                  %cluster info of sup2 from fnm
C = ~isempty(intersect(xC1,xC2));                  %if they intersect?
```

SupCSup2D evaluates whether the two given supports are connected to each other or not. This is a pretty and robust implementation of connectivity assessment between the two supports. Note that the cluster information of fracture network (*La*) is used for the evaluation.

2.4.14 SupXLines2D

```
% SupXLines2D
% find cluster labels for a support intersecting 2D fracture network
%
% Usage :
%    xC = SupXLines2D(sup,lines,La)
%
% input : sup       (4), e.g., [0,1,0,1]
%         lines     (n,4)
%         La        cluster labels of fractures
% output: xC        intersected clusters
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function xC = SupXLines2D(sup,lines,La)
xC = unique(La(sum(ClipLines2D(lines,sup),2)>0));   %utilises clipping concept
```

SupXLines2D is used to determine all fracture clusters (including isolated fractures) intersecting a support.

2.4.15 SupXNLines2D

```
% SupXNLines2D
% finds number of intersected 2D fractures by a 2D support
%
% Usage :
%    xN = SupXNLines2D(sup,lines)
%
% input : sup       (4), e.g., [0,1,0,1]
%         lines     (n,4)
% output: xN        number of intersected lines
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function xN = SupXNLines2D(sup,lines)
xN = sum(sum(ClipLines2D(lines,sup),2)>0);        %utilises clipping concept
```

SupXNLines2D determines the total number of intersecting fractures for a support.

2.4.16 P21G

```
% P21G
% evaluates P21 measure of 2D fracture network (regular grid)
%
% Usage :
%    [tLs,xLs,x,y] = P21G(lines,gn,gm)
%
% input : lines     (n,4)
%         gn        grid dimensions, horizontally
%         gm        grid dimensions, vertically
% output: tLs       (n) cell, total length
%         xLs       (n) cell, all lengths
%         x,y       extent for plotting
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [tLs,xLs,x,y] = P21G(lines,gn,gm)
if nargin<3
    gn = 25;
    gm = gn;
end
w = 1/gn;                                         %cell width
h = 1/gm;                                         %cell height
xLs = cell(gm,gn);
tLs = zeros(gm,gn);
for i = 1:gm
    for j = 1:gn
        sup = [(j-1)*w,j*w,(i-1)*h,i*h];
        cls = ClipLines2D(lines,sup);
        lhs = Lengths2D(cls(sum(cls,2)>0,:));
```

```
        xLs(i,j) = {lhs};                          %store all lengths
        tLs(i,j) = sum(lhs);                       %total length per cell
    end
end
x = [w/2,1-w/2];
y = [h/2,1-h/2];
```

P21G is to compute **P21** measure based on regular grid cell sampling.

2.4.17 ConnectivityIndex2D

```
% ConnectivityIndex2D
% computes connectivity index (CI) on 2D fracture networks
%
% Usage :
%    [CI,x,y] = ConnectivityIndex2D(lines,La,gm,gn,cm,cn)
%
% input : lines     (n,4)
%         La        cluster labels (n)
%         gm        grid dimension vertically
%         gn        grid dimension horizontally
%         cm        target cell i index
%         cn        target cell j index
% output: CI        (gm,gn)
%         x,y       extents
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [CI,x,y] = ConnectivityIndex2D(lines,La,gm,gn,cm,cn)
w = 1/gn;
h = 1/gm;
CI = zeros(gm,gn);
xCs = cell(gm,gn);
for i = 1:gm                                    %extract cluster info for all supports
    for j = 1:gn
        sup = [(j-1)*w,j*w,(i-1)*h,i*h];
        xCs(i,j) = {SupXLines2D(sup,lines,La)};   %store
    end
end
xC1 = xCs{cm,cn};                               %target cell cluster info
if ~isempty(xC1)                                %if target cell is not isolated
    for i = 1:gm
        for j = 1:gn
            com = intersect(xC1,xCs{i,j});      %any common cluster?
            if ~isempty(com); CI(i,j) = 1; end
        end
    end
end
x = [w/2,1-w/2];                                %for usage in imagesc(x,y,CI)
y = [h/2,1-h/2];
```

ConnectivityIndex2D evaluates the connectivity index (**CI**) on a two-dimensional fracture network based on grid cell sampling.

2.4.18 ConnectivityField2D

```
% ConnectivityField2D
% computes connectivity field (CF) for 2D fracture network
%
% Usage :
%    [CF,x,y] = ConnectivityField2D(lines,La,gm,gn,rm,rn)
%
% input : lines     (n,4)
%         La        cluster labels (n)
%         gm        grid dimension vertically
%         gn        grid dimension horizontally
%         rm        range for cells vert.
%         rn        range for cells horiz.
% output: CF        (gm,gn)
%         x,y       extents
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function [CF,x,y] = ConnectivityField2D(lines,La,gm,gn,rm,rn)
w = 1/gn;
h = 1/gm;
sm = length(rm);
sn = length(rn);
CF = zeros(gm,gn);
xCs = cell(gm,gn);
for i = 1:gm
    for j = 1:gn
        sup = [(j-1)*w,j*w,(i-1)*h,i*h];
        xCs(i,j) = {SupXLines2D(sup,lines,La)};   %all supports' clusters information
    end
end
for i = 1:sm                                      %outer loop for all target cells
    for j = 1:sn
        xC1 = xCs{i,j};
        k = 0;
        for ii = 1:gm
            for jj = 1:gn
                com = intersect(xC1,xCs{ii,jj});
                if ~isempty(com); k = k+1; end    %record number of connected cells
            end
        end
        CF(rm(i),rn(j)) = k;
    end
end
x = [w/2,1-w/2];                                  %for usage in imagesc(x,y,CF)
y = [h/2,1-h/2];
```

ConnectivityField2D computes connectivity field (**CF**) on a two-dimensional fracture network based on grid cell sampling.

2.4.19 BreakLinesX2D

```
% BreakLinesX2D
% breaks 2D lines at their intersection points
%
% Usage :
```

```
%    olins = BreakLinesX2D(lines)
%
% input : lines     (n,4)
% output: olins     cell
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function olins = BreakLinesX2D(lines)
[xts,ids,~] = LinesX2D(lines);
gxs = Group(xts,ids,size(lines,1));
n = size(gxs,1);
olins = cell(n,1);
for i=1:n
    ots = SortPoints2D([gxs{i};lines(i,1:2);lines(i,3:4)]);
    olins{i} = [ots(1:end-1,:),ots(2:end,:)];
end
```

BreakLinesX2D breaks given lines at their intersection points. This function is used for generating backbone structure.

2.4.20 Rotate2D

```
% Rotate2D
% rotates 2D points about a center by given angle
%
% Usage :
%    ots = Rotate2D(pts,cnt,ang)
%
% input : pts       (n,2)
%         cnt       center of rotation (2)
%         ang       angle of rotation
% output: ots       (n,2)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function ots = Rotate2D(pts,cnt,ang)
if nargin<3; ang = 0; end
if nargin<2; cnt = [0,0]; end
ots = [pts(:,1)-cnt(1),pts(:,2)-cnt(2)]*[cos(ang),sin(ang);-sin(ang),cos(ang)];
ots = [ots(:,1)+cnt(1),ots(:,2)+cnt(2)];
```

Rotate2D applies rotation to points about a given centre and angle (in radian).

2.4.21 SortPoints2D

```
% SortPoints2D
% sorts 2D points topologically
%
% Usage:
%    ots = SortPoints2D(pts)
%
% input : pts       (n,2)
% output: ots       (n,2)
```

```
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function ots = SortPoints2D(pts)
p1 = FarthestPoints(pts);                          %farthest point as ref
d = pdist2(pts,p1);
[~,idx] = sort(d);
ots = pts(idx,:);
```

SortPoints2D sorts points based on their topological arrangement, that is, they can be connected to create a polyline without self-intersection.

2.4.22 Isolated2D

```
% Isolated2D
% checks if a line at index i from lines is isolated
%
% Usage :
%    b = Isolated2D(i,lines,tol)
%
% input : i         index
%         lines     (n,4)
%         tol       tolerance, default=1e-9
% output: b         boolean
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function b = Isolated2D(i,lines,tol)
if nargin<3; tol = 1e-9; end
p1s = lines(:,1:2);
p2s = lines(:,3:4);
pts = [p1s;p2s];
b = (Occurrence(p1s(i,:),pts,tol)<=1) | (Occurrence(p2s(i,:),pts,tol)<=1);
```

Isolated2D determines if a line is isolated from (unconnected to) others in fracture network.

2.4.23 Backbone2D

```
% Backbone2D
% returns backbone of 2D fracture network
%
% Usage :
%    bbn = Backbone2D(lines,process,rgn,tol)
%
% input : lines     (n,4)
%         process   if true to break lines into segments
%         rgn       the region of study as polyline (k,4)
%         tol       tolerance
% output: bbn       (m,4)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
```

```
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function bbn = Backbone2D(lines,process,rgn,tol)
if nargin<4; tol = 1e-9; end
if (nargin<3) || isempty(rgn)
    rgn = [0,0,1,0; 1,0,1,1; 1,1,0,1; 0,1,0,0];   %default: a unit square
end
if nargin<2; process = false; end
if process
    bbn = Stack(BreakLinesX2D([lines;rgn]));
else
    bbn = lines;
end
while true
    B = IsolatedLines2D(bbn,tol);
    if ~any(B); break; end                         %break if no isolated line anymore
    bbn = bbn(~B,:);                               %update backbone
end
```

Backbone2D extracts the backbone structure of given fracture network.

2.4.24 IsolatedLines2D

```
% IsolatedLines2D
% check isolation for all 2D fracture lines
%
% Usage :
%    B = IsolatedLines2D(lines,tol)
%
% input : lines     (n,4)
%         tol
% output: B         (n) boolean
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function B = IsolatedLines2D(lines,tol)
if nargin<3; tol = 1e-9; end
p1s = lines(:,1:2);
p2s = lines(:,3:4);
pts = [p1s;p2s];
n = size(lines,1);
B = false(n,1);
for i=1:n
    B(i) = (Occurrence(p1s(i,:),pts,tol)<=1) | (Occurrence(p2s(i,:),pts,tol)<=1);
end
```

IsolatedLines2D checks isolation for all lines in fracture network.

2.4.25 BackboneToNodesEdges2D

```
% BackboneToNodesEdges2D
% returns (nodes,edges) extracted from 2D backbone
%
```

```
% Usage :
%    [nodes,edges] = BackboneToNodesEdges2D(bbn)
%
% input : bbn       backbone
% output: nodes
%         edges
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [nodes,edges] = BackboneToNodesEdges2D(bbn)
nodes = dict();
edges = bbn;
for i=1:size(bbn,1)
    p1 = bbn(i,1:2);                    %endpoints as keys for nodes
    p2 = bbn(i,3:4);
    nodes(p1) = unique([nodes(p1),i]);   %indices of edges associated with nodes
    nodes(p2) = unique([nodes(p2),i]);
end
```

BackboneToNodesEdges2D produces list of nodes and edges (*graph structure*) from given backbone.

2.4.26 Expand2D

```
% Expand2D
% expands a 2D matrix by nx,ny
%
% Usage :
%    Y = Expand2D(X,nx,ny)
%
% input : X         2D matrix
%         nx,ny     increase in dimensions
% output: Y         expanded matrix
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function Y = Expand2D(X,nx,ny)
if nargin==1; nx = 1; ny = 1; end
[m,n] = size(X);
Y = zeros(m+ny,n+nx)+X(end,end);
Y(1:m,1:n) = X;
```

Expand2D expands given 2D matrix by *nx* and *ny*.

2.4.27 Resize2D

```
% Resize2D
% resizes 2D matrix in shape (m,n)
%
% Usage :
%    B = Resize2D(A,m,n)
%
% input : A         input 2d matrix
```

```
%         m,n       desired dimensions
% output: B         resized matrix (m,n)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function B = Resize2D(A,m,n)
[a,b] = size(A);
if nargin<2; m = 70; end
if nargin<3; n = m; end
zm = m/a;
zn = n/b;
B = A(floor((0:end*zm-1)/zm)+1,floor((0:end*zn-1)/zn)+1);
```

Resize2D resizes a given matrix into shape of (m,n).

2.4.28 DrawLines2D

```
% DrawLines2D
% draws quickly 2D fracture lines
%
% Usage :
%    DrawLines2D(lines,La,rgb,age)
%
% input : lines     (n,4)
%         La        cluster labels
%         rgb       color, default=[0,0,0]
%         age       axes, grid settings, check `Titles2D`
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function DrawLines2D(lines,La,rgb,age)
if nargin<4; age = '=['; end
if nargin<3; rgb = [0,0,0]; end
if nargin<2; La = []; end
if isempty(La) || all(La==0)
    [X,Y] = LinesToXYnan2D(lines);
    if all(rgb==0); rgb = [0,0,0]; end
    plot(X,Y,'-','Color',rgb)
else
    [X,Y] = LinesToXYnan2D(lines(La<0,:));        %isolated fractures
    plot(X,Y,'-','Color',[0.5,0.5,0.5])
    hold on
    for i=1:max(La)                               %fracture clusters
        [X,Y] = LinesToXYnan2D(lines(La==i,:));
        plot(X,Y,'-','Color',rand(3,1),'LineWidth',1.5);
    end
end
Titles2D(age)
```

DrawLines2D draws efficiently large number of fracture lines.

2.4.29 LinesToXYnan2D

```
% LinesToXYnan2D
% builds [X,Y,nan] from lines for `plot` to highest efficiency
%
% Usage :
%    [X,Y] = LinesToXYnan2D(lines)
%
% input : lines     (n,4)
% output: X         x coordinates of lines
%         Y         y coordinates of lines
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [X,Y] = LinesToXYnan2D(lines)
n = size(lines,1);
X = [lines(:,[1,3]),NaN(n,1)]';
Y = [lines(:,[2,4]),NaN(n,1)]';
```

LinesToXYnan2D builds $(data, NaN)$ structure for increased efficiency in drawing large number of 2D lines.

2.4.30 ExpandAxes2D

```
% ExpandAxes2D
% expands current axes by rate (+:relative,-:absolute)
%
% Usage:
%    ExpandAxes2D(rate)
%
% input : rate      to expand axes symmetrically
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function ExpandAxes2D(rate)
a = axis;
if rate>0                                          %relative expansion
    fx = a(2)-a(1);
    fy = a(4)-a(3);
else                                               %absolute expansion
    fx = -1;
    fy = -1;
end
axis([a(1)-rate*fx,a(2)+rate*fx,a(3)-rate*fy,a(4)+rate*fy]);
```

ExpandAxes2D expands two-dimensional axes by factor of given rate.

2.4.31 Titles2D

```
% Titles2D
% sets titles, grid etc for current 2D axes
%
```

```
% Usage :
%    Titles2D(tl,xl,yl,ext,rgn,age)
%
% input : tl        title
%         xl        xlabel
%         yl        ylabel
%         ext       limits of axes
%         rgn       region of study
%         age       aspect,limits,grid switches
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function Titles2D(age,tl,xl,yl,ext,rgn)
if nargin<6; rgn = [0,1,0,1]; end
if nargin<5; ext = [-0.03,1.03,-0.03,1.03]; end
if nargin<4; yl = 'Y'; end
if nargin<3; xl = 'X'; end
if nargin<2; tl = ''; end
if nargin<1; age = '=['; end
hold on
drawBox(rgn,'k-','LineWidth',1.5)
if isempty(strfind(age,'-'))
    title(tl)
    xlabel(xl);
    ylabel(yl);
else
    axis off
end
box on
if strfind(age,'=')>0; axis image; end
if strfind(age,'+')>0; grid on; end
if strfind(age,'[')>0; axis(ext); end
hold off
```

Titles2D simplifies setting the titles, labels, grid, extent and region information for current 2D axes.

2.5 Functions for Three Dimensional Fracture Networks

2.5.1 RandPoly3D

```
% RandPoly3D
% generates randomly shaped and distributed 3D polygons
%
% Usage :
%    plys = RandPoly3D(n,dax,day,daz)
%
% input : n         number of polygons
%         dax,day,daz rotation angle range around X, Y and Z axes
% output: ply       (n,4,3)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
```

```
function plys = RandPoly3D(n,dax,day,daz)
if nargin<4; daz = 2*pi; end
if nargin<3; day = 2*pi; end
if nargin<2; dax = 2*pi; end
if nargin<1; n = 1; end
plys = zeros(n,4,3);
for i=1:n
    r = rand(4,1);
    ply = [r(1),0,0; 0,r(2),0; r(3),1,0; 1,r(4),0];   %random polygon with 4 vertices
    cnt = polygonCentroid3d(ply);
    T = composeTransforms3d(...
        createRotationOx(cnt,rand*dax),...
        createRotationOy(cnt,rand*day),...
        createRotationOz(cnt,rand*daz));
    plys(i,:,:) = transformPoint3d(ply(:,1),ply(:,2),ply(:,3),T);
end
```

RandPoly3D generates three-dimensional polygons following the given angle variations about triple axes.

2.5.2 GenFNM3D

```
% GenFNM3D
% generates 3D fracture network
%
% Usage:
%    [clys,plys] = GenFNM3D(n,dax,day,daz,s,rgn)
%
% input : n         number of polygons
%         dax,day,daz rotation angle range around X, Y and Z axes
%         s         scale
%         rgn       region of study, default= unit cube
% output: clys      (n), cell, clipped polygons by region of study rgn
%         plys      (n), cell
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [clys,plys] = GenFNM3D(n,dax,day,daz,s,rgn)
if nargin<6; rgn = [0,1,0,1,0,1]; end
if nargin<5; s = 0.25; end
if nargin<4; daz = pi; end
if nargin<3; day = pi; end
if nargin<2; dax = pi; end
if nargin<1; n = 1; end
plys = cell(n,1);
for i=1:n
    ply = [rand-0.5,-0.5,0; -0.5,rand-0.5,0; rand-0.5,0.5,0; 0.5,rand-0.5,0];
    pt = rand(3,3);
    T = composeTransforms3d(...
        createRotationOx([0,0,0],(2*rand-1)*dax),...
        createRotationOy([0,0,0],(2*rand-1)*day),...
        createRotationOz([0,0,0],(2*rand-1)*daz),...
        createScaling3d(s,s,s),...
        createTranslation3d(pt(1),pt(2),pt(3)));
    plys{i} = transformPoint3d(ply(:,1),ply(:,2),ply(:,3),T);
end
clys = ClipPolys3D(plys,rgn);
```

GenFNM3D synthesises a three-dimensional fracture network (polygonal shape fractures).

2.5.3 Sup3D

```
% Sup3D
% creates a 3D support (box)
%
% Usage:
%    sup = Sup3D(cnt,dim)
%
% input : cnt       a point(3)
%         dim       [width,height,depth]
% output: sup       cell, six sides of a cube
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function sup = Sup3D(cnt,dim)
if nargin<2; dim = [1,1,1]; end
if nargin<1; cnt = [0.5,0.5,0.5]; end
sup = cell(6,1);
sup{1} = [0,0,0; 0,0,1; 0,1,1; 0,1,0];    %left
sup{2} = [1,0,0; 1,0,1; 1,1,1; 1,1,0];    %right
sup{3} = [0,0,0; 1,0,0; 1,1,0; 0,1,0];    %bottom
sup{4} = [0,0,1; 1,0,1; 1,1,1; 0,1,1];    %top
sup{5} = [0,0,0; 0,0,1; 1,0,1; 1,0,0];    %front
sup{6} = [0,1,0; 0,1,1; 1,1,1; 1,1,0];    %back
d = cnt-[0.5*dim(1),0.5*dim(2),0.5*dim(3)];
for i=1:6
    sup{i}(:,1) = sup{i}(:,1)*dim(1)+d(1);
    sup{i}(:,2) = sup{i}(:,2)*dim(2)+d(2);
    sup{i}(:,3) = sup{i}(:,3)*dim(3)+d(3);
end
```

Sup3D creates 3D cubic support.

2.5.4 ClipPolys3D

```
% ClipPolys3D
% clips 3D polygons by a box
%
% Usage :
%    clys = ClipPolys3D(plys,box)
%
% input : plys      cell(n)
% output: clys      cell(m)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function clys = ClipPolys3D(plys,box)
if nargin<2; box = [0,1,0,1,0,1]; end
x1 = box(1); x2 = box(2); y1 = box(3);
y2 = box(4); z1 = box(5); z2 = box(6);
```

```
xy1 = createPlane([x1,y1,z1],[0,0,-1]);
xy2 = createPlane([x1,y1,z2],[0,0,1]);
xz1 = createPlane([x1,y1,z1],[0,-1,0]);
xz2 = createPlane([x1,y2,z1],[0,1,0]);
yz1 = createPlane([x1,y1,z1],[-1,0,0]);
yz2 = createPlane([x2,y1,z1],[1,0,0]);
n = size(plys,1);
clys = cell(n,1);
for i=1:n
    ply = plys{i};
    ply = clipConvexPolygon3dHP(ply,xy1);
    ply = clipConvexPolygon3dHP(ply,xy2);
    ply = clipConvexPolygon3dHP(ply,xz1);
    ply = clipConvexPolygon3dHP(ply,xz2);
    ply = clipConvexPolygon3dHP(ply,yz1);
    ply = clipConvexPolygon3dHP(ply,yz2);
    if all(ply(1,:)==ply(end,:)); ply = ply(1:end-1,:); end
    clys{i} = ply;
end
```

ClipPolys3D clips 3D polygons by given cube.

2.5.5 PolysX3D

```
% PolysX3D
% finds all intersections between 3D polygons
%
% Usage:
%    [xts,ids,La] = PolysX3D(plys)
%
% input : plys      cell(n)
% output: xts       cell
%         ids       cell
%         La        cluster labels
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [xts,ids,La] = PolysX3D(plys)
n = size(plys,1);
m = n*(n-1)/2;
xts = cell(m,1);
ids = cell(m,1);
k = 0;
for i = 1:n-1
    for j = i+1:n
        xpt = PolyXPoly3D(plys{i},plys{j});
        if isempty(xpt); continue; end
        k = k+1;
        xts{k} = xpt;                           %intersection points
        ids{k} = int32([i,j]);                  %intersecting lines indices
    end
end
xts = xts(1:k);
ids = ids(1:k);
La = Labels(Clusters(ids),n);                   %fracture cluster labels
```

PolysX3D conducts intersection analysis for given 3D fractures resulting in information of intersection points, inter-connected fracture indices and fracture clusters.

2.5.6 PolysXPolys3D

```
% PolysXPolys3D
% finds intersection (points,indices) between two sets of 3D polygons
%
% Usage:
%    [xts,pts,ids] = PolysXPolys3D(plys1,plys2)
%
% input : plys1     cell of polygons
%         plys2     cell of polygons
% output: xts       (k,3), cell
%         pts       stacked intersection points, i.e., (k,2)
%         ids       indices, cell
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [xts,pts,ids] = PolysXPolys3D(plys1,plys2)
m = length(plys1);
n = length(plys2);
xts = cell(m,1);
ids = [];
u = 0;
for i=1:m
    pts = cell(n,1);
    idx = zeros(n,1);
    k = 0;
    for j=1:n
        xpt = PolyXPoly3D(plys1{i},plys2{j});     %intersecting
        if isempty(xpt); continue; end
        k = k+1;
        pts{k} = xpt;                             %points
        idx(k) = j;                               %polygon index
    end
    if k==0; continue; end
    u = u+1;
    xts{u} = pts(1:k);
    ids = union(ids,idx(1:k));                    %indices
end
xts = xts(1:u);
pts = zeros(0,3);
k = 0;
for i=1:u
    cps = xts{i};                                 %stacking all intersection points
    for j=1:size(cps,1)
        ets = cps{j};
        for w=1:size(ets,1)
            k = k+1;
            pts(k,:) = ets(w,:);
        end
    end
end
```

PolysXPolys3D assesses inter-connection between two sets of three-dimensional fractures. It can be used for example to find connectivity information between a three-dimensional support (e.g., cube) and a fracture network.

2.5.7 PolyXPoly3D

```
% PolyXPoly3D
% finds intersection points between two 3D polygons
%
% Usage :
%    xts = PolyXPoly3D(ply1,ply2)
%
% input : ply1      (n,3)
%         ply2      (m,3)
% output: xts       (k,3)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function xts = PolyXPoly3D(ply1,ply2)
edges = [ply1,circshift(ply1,[-1,0])];            %create edges
pln = createPlane(ply2(1:3,:));
xts = intersectEdgePlane(edges,pln);
xts = xts(sum(isnan(xts),2)==0,:);
if ~isempty(xts)
    pts = planePosition(ply2,pln);
    its = planePosition(xts,pln);
    ins = xor(isPointInPolygon(its,pts),polygonArea(pts)<0);
    xts = xts(ins,:);
end
```

PolyXPoly3D results in intersection points between two three-dimensional polygons.

2.5.8 SupCSup3D

```
% SupCSup3D
% checks if two 3D suports are connected
%
% Usage :
%    out = SupCSup3D(sup1,sup2,plys,La)
%
% input : sup1      cell of polygons
%         sup2      cell of polygons
%         plys      cell of polygons
%         La        cluster labels
% output: out       boolean
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function out = SupCSup3D(sup1,sup2,plys,La)
[~,~,ids1] = PolysXPolys3D(sup1,plys);            %fractures intersection indices
```

```
[~,~,ids2] = PolysXPolys3D(sup2,plys);
if isempty(ids1) || isempty(ids2)
    out = false;
else
    out = ~isempty(intersect(La(ids1),La(ids2)));
end
```

SupCSup3D evaluates connectivity between two three-dimensional supports (e.g., cubes).

2.5.9 BBox3D

```
% BBox3D
% finds min and max of polygon in each axis X, Y and Z
%
% Usage :
%    [mins,maxs] = BBox3D(plys)
%
% input : plys      cell
% output: mins      min values for X, Y and Z
%         maxs      max values for X, Y and Z
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [mins,maxs] = BBox3D(plys)
mins = min(cell2mat(cellfun(@min,plys,'UniformOutput',false)));
maxs = max(cell2mat(cellfun(@max,plys,'UniformOutput',false)));
```

BBox3D finds bounding box for given three-dimensional points / polygons.

2.5.10 Expand3D

```
% Expand3D
% expands a 3D matrix by nx,ny,nz
%
% Usage :
%    Y = Expand3D(X,nx,ny,nz)
%
% input : X         3D matrix
%         nx,ny,nz  increase in dimensions
% output: Y         expanded matrix
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function Y = Expand3D(X,nx,ny,nz)
if nargin==1; nx = 1; ny = 1; nz = 1; end
[m,n,o] = size(X);
Y = zeros(m+ny,n+nx,o+nz)+X(end,end,end);
Y(1:m,1:n,1:o) = X;
```

Expand3D expands a given three-dimensional matrix.

2.5.11 Resize3D

```
% Resize3D
% resizes 3D matrix into shape (m,n,o)
%
% Usage :
%    B = Resize3D(A,m,n,o)
%
% input : A         input 3d matrix
%         m,n,o     desired dimensions
% output: B         resized matrix (m,n,o)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function B = Resize3D(A,m,n,o)
[a,b,c] = size(A);
if nargin<2; m = 70; end
if nargin<3; n = m; end
if nargin<4; o = n; end
zm = m/a;
zn = n/b;
zo = o/c;
B = A(floor((0:end*zm-1)/zm)+1,floor((0:end*zn-1)/zn)+1,floor((0:end*zo-1)/zo)+1);
```

Resize3D resizes three-dimensional matrix to a given shape. This function is useful to combine different shapes of input matrices to generate E-Type maps, for example.

2.5.12 SaveToFile3D

```
% SaveToFile3D
% saves 3D data into text file
%
% Usage :
%    SaveToFile3D(fname,x)
%
% input : fname     filename
%         x         3D matrix
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function SaveToFile3D(fname,x,index)
if nargin<3; index = false; end
if ~index
    dlmwrite(fname,x,'delimiter',',')            %just data, no indices
else
    fut = fopen(fname,'w');
    [m,n,o] = size(x);
    for i=1:m
        for j=1:n
            for k=1:o
                fprintf(fut,sprintf('%d, %d, %d, %0.6f\n',i,j,k,x(i,j,k)));
            end
```

```
        end
    end
    fclose(fut);
end
```

SaveToFile3D exports three-dimensional data to an *ASCII* file.

2.5.13 SavePolysToVTK3D

```
% SavePolysToVTK3D
% saves 3D polygons as standard VTK file (ASCII format)
%
% Usage :
%    SavePolysToVTK3D(plys,colors,fname)
%
% input : plys      cell
%         colors    (x,3[4]), with or without alpha values
%         fname     filename
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function SavePolysToVTK3D(plys,colors,fname)
nply = length(plys);
plyn = cellfun(@length,plys,'UniformOutput',false);
npnt = sum(cell2mat(plyn));
fut = fopen(fname,'w');
fprintf(fut,'# vtk DataFile Version 3.0\n');      %header...
fprintf(fut,'Polygons by Younes Fadakar Alghalandis\n');
fprintf(fut,sprintf('ASCII\nDATASET POLYDATA\nPOINTS %d float\n',npnt));
for i=1:nply
    ply = plys{i};
    fprintf(fut,sprintf('%0.6f %0.6f %0.6f\n',ply'));
end
fprintf(fut,sprintf('POLYGONS %d %d\n',nply,npnt+nply));
k = 0;
for i=1:nply                                      %polygon data
    fprintf(fut,strcat(sprintf('%d ',plyn{i},k:k+plyn{i}-1),'\n'));
    k = k+plyn{i};
end
fprintf(fut,sprintf('POINT_DATA %d\n',npnt));
fprintf(fut,'COLOR_SCALARS lut 4\n');
if size(colors,2)==3                              %if no alpha provided, set all to 1
    colors(:,4) = 1;
end
for i=1:nply
    for j=1:plyn{i}
        fprintf(fut,strcat(sprintf('%0.3f ',colors(i,:)),'\n'));
    end
end
fclose(fut);
fprintf(1,'Polygons were saved as file %s.\n',fname);   %report on screen
```

SavePolysToVTK3D exports polygons to a file as **VTK** ASCII format. VTK format is industry standard for three-dimensional data and can be visualised and manipulated by many available free software applications such as ***ParaView***.

2.5.14 SetAxes3D

```
% SetAxes3D
% sets and adjust axes into 3D view
%
% Usage :
%    SetAxes3D(mins,maxs)
%
% input : mins,maxs min and max values of X, Y and Z axes
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function SetAxes3D(mins,maxs)
if nargin==0;
    mins = [0,0,0];                                %defaults
    maxs = [1,1,1];
end
hold on
plot3([mins(1),maxs(1)],[mins(2),mins(2)],[mins(3),mins(3)],'-',...
      'LineWidth',1.5,'Color',[0.7,0,0]);
plot3([mins(1),mins(1)],[mins(2),maxs(2)],[mins(3),mins(3)],'-',...
      'LineWidth',1.5,'Color',[0,0.7,0]);
plot3([mins(1),mins(1)],[mins(2),mins(2)],[mins(3),maxs(3)],'-',...
      'LineWidth',1.5,'Color',[0,0,0.7]);
text(0.5*(mins(1)+maxs(1)),mins(2),mins(3),'X','BackgroundColor',[0.7,0,0],'Color','w
')
text(mins(1),0.5*(mins(2)+maxs(2)),mins(3),'Y','BackgroundColor',[0,0.7,0],'Color','w
')
text(mins(1),mins(2),0.5*(mins(3)+maxs(3)),'Z','BackgroundColor',[0,0,0.7],'Color','w
')
camproj('perspective')
set(gca,'CameraPosition',[-1*maxs(1),-2*maxs(2),1.5*maxs(3)]);
axis(reshape([mins;maxs],1,[]));
axis image
grid on
box on
```

SetAxes3D sets the current view into three-dimensional perspective view with axes and labels automatically adjusted.

2.5.15 DrawPolys3D

```
% DrawPolys3D
% draws quickly 3D polygons
%
% Usage :
%    DrawPolys3D(plys,La,rgba,axes)
%
% input : plys      cell
%         rgba      [r,g,b,a]
%         clus      La
%         axes      if true to draw set axes and adjust into 3D
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
```

```
% Updated: Nov 2013
function DrawPolys3D(plys,La,rgba,axes)
if nargin<4; axes = true; end
if nargin<3; rgba = [0.5,0,0.1,0.5]; end
if nargin<2; La = []; end
hold on
cmap = colormap(jet);
for i=1:length(plys)
    ply = plys{i};
    if ~isempty(La) && La(i)<0                    %isolated fractures
        patch(ply(:,1),ply(:,2),ply(:,3),[0.5,0.5,0.5],'FaceAlpha',0.5,...
              'EdgeColor','none');
    else
        fvc = zeros(length(ply),3);
        if isempty(La)                            %no cluster labels info provided
            fvc(1,:) = rgba(1:3);
        else
            fvc(1,:) = cmap(int32(double(La(i))/double(max(La))*64),:);
        end
        h = patch(ply(:,1),ply(:,2),ply(:,3),0,'FaceAlpha',rgba(4));
        set(h,'FaceVertexCData',fvc);
    end
end
if axes
    [mins,maxs] = BBox3D(plys);                   %bounding box of polygons
    SetAxes3D(mins,maxs);
end
```

DrawPolys3D draws three-dimensional polygons (fracture network). If clusters' labels (*La*) were provided fractures will be colourised according to their associated clusters.

2.5.16 DrawSlices3D

```
% DrawSlices3D
% draws 3D slices of 3D volume data
%
% Usage :
%    h = DrawSlices3D(data,a,axes)
%
% input : data      3D array
%         a         transparency
%         axes      if true to set and adjust axes into 3D
% output: h         handle to slice objects
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function h = DrawSlices3D(data,a,axes)
if nargin<3; axes = true; end
if nargin<2; a = 1; end
[m,n,o] = size(data);
[x,y,z] = meshgrid(0:m,0:n,0:o);
h = slice(x,y,z,Expand3D(data),m/2,n/2,o/2);
shading flat
if a~=1
    if a<0
```

```
        set(h,'EdgeColor','none','FaceColor','interp');
        alpha(abs(a));
    else
        for i=1:length(h)
            set(h(i),'alphadata',get(h(i),'cdata'),'facealpha',a);
        end
    end
end
if nargout==0; clear h; end
if axes; SetAxes3D([m,n,o]); end
```

DrawSlices3D draws three-dimensional slices on the middle of each of axes.

2.5.17 VolRender3D

```
% VolRender3D
% view volume render of 3D volumetric data
%
% Usage :
%    VolRender3D(data,a,axes)
%
% input : data      array (m,n,o)
%         a         alpha factor
%         axes      if true to draw set axes and adjust into 3D
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function VolRender3D(data,a,axes)
if nargin<3; axes = true; end
if nargin<2; a = 1; end
[m,n,o] = size(data);
mdl = Vol3D('CData',data);
alphamap('rampup');
alphamap(a.*alphamap);
if axes; SetAxes3D([0,0,0],[m,n,o]); end
```

VolRender3D renders three-dimensional data (volumetric) with adjustable alpha (transparency value).

2.5.18 Vol3D

```
function [model] = vol3d(varargin)                    → Vol3D
By Woodford O, 2011
```

2.6 Generic Functions

2.6.1 Scale

```
% Scale
% scales (maps) X into range (a to b)
%
% Usage :
%    Y = Scale(X,a,b)
%
% input : X         any array
%         a         minimum bound of the output
%         b         maxmium bound of the output
% output: Y         same as X but mapped
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function Y = Scale(X,a,b)
if nargin<3; a = 0; b = 1; end
Y = double(X-min(X(:)))/double(range(X(:)))*(b-a)+a;
```

Scale scales data to a given bounds.

2.6.2 ToStruct

```
% ToStruct
% builds `struct` data type from data
%
% Usage :
%    S = ToStruct(data)
%
% input : data      any array
% output: S         struct
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function S = ToStruct(data)
S = struct();
m = size(data,1);
for i = 1:m
   S.(sprintf('S%d',i)) = data(i,:);
end
```

ToStruct provides "*struct*" format for the given data.

2.6.3 KDE

```
function [bandwidth,density,X,Y] = kde2d(data,n,MIN_XY,MAX_XY)    → KDE
By Botev Z.I @ botev@maths.uq.edu.au
```

KDE applies kernel density estimation.

2.6.4 Smooth

```
function [z,s,exitflag,Wtot] = smoothn(varargin)                    → Smooth
By Garcia D @ http://www.biomecardio.com/matlab/smoothn.html
```

Smooth applies smoothing on the given data.

2.6.5 dict

```
classdef dict < handle                                               → dict
By Harriman D @ doug.harriman@gmail.com
```

dict provides "*dict*" structure.

2.6.6 Clusters

```
% Clusters
% clusters items based on common elements
%
% Usage :
%    C = Clusters(S)
%
% input : S         cell
% output: C         cell
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function C = Clusters(S)
if isempty(S); C = {}; return; end
while true
    m = length(S);
    united = zeros(m,1);                          %nothing is clustered yet
    C = cell(m,1);
    u = 0;
    for i = 1:m-1
        if united(i); continue; end;
        p = S{i};
        for j = i+1:m
            q = S{j};
            com = intersect(p,q);                 %common elements
            if ~isempty(com)
                united(j) = 1;
                p = union(p,q);
                S{i} = p;
            end
        end
        u = u+1;
        C{u} = p;
    end
    if ~united(m)                                 %copy the last item if not united
        u = u+1;
        C{u} = S{m};
```

```
    end
    C = C(1:u);
    if any(united)
        S = C;
    else                                        %all are united, i.e., clustered
        C = S;
        break
    end
end
```

Clusters determines fracture clusters in the fracture network by means of intersection indices (see also **LinesX2D**). This function is highly efficient and also generic for two- and three-dimensional fracture clustering.

2.6.7 CheckClusters

```
% CheckClusters
% checks if clusters are OK, have no missing common element
%
% Usage :
%    OK = CheckClusters(C)
%
% input : C         cell
% output: OK        boolean
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function OK = CheckClusters(C)
X = horzcat(C{:});
OK = (length(unique(X))==length(X));
```

CheckClusters checks cluster information for any inconsistency due to any remaining unclassified elements.

2.6.8 Labels

```
% Labels
% extracts labels from clusters
%
% Usage :
%    La = Labels(C,n)
%
% input : C         cell of Clusters
%         n         number of fractures
% output: La        labels for all fractures
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function La = Labels(C,n)
La = zeros(n,1);
```

```
for i = 1:length(C)
    La(C{i}) = i;
end
f = (La==0);                                    %isolated fractures
La(f) = -(1:sum(f));                            %relabeling
```

Labels assigns unique label for each cluster. Isolated fractures are assigned a unique negative label for each.

2.6.9 Relabel

```
% Relabel
% relabel cluster labels according to their number of elements
%
% Usage :
%    Ra = Relabel(La)
%
% input : La        labels
% output: Ra        relabeled output
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function Ra = Relabel(La)
k = max(La);                                    %highest label
frq = zeros(k,1,'int32');
for i=1:k                                       %frequency of each label
    frq(i) = sum(La==i);
end
[~,idx] = sort(frq);                            %sort based on their frequencies
Ra = La;
for i=1:k
    Ra(La==idx(i)) = i;                         %apply relabing
end
```

Relabel is to sort fracture cluster labels based on the cardinality of each cluster.

2.6.10 Stack

```
% Stack
% stacks values of cell, i.e., results in array
%
% Usage :
%    S = Stack(C)
%
% input : C         cell
% output: S         array
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function S = Stack(C)
S = cat(1,C{:});
```

Stack stacks data in the give cell structure and produces an array.

2.6.11 Group

```
% Group
% groups intersection indices and points
%
% Usage :
%    [gxs,gds] = Group(xts,ids,n)
%
% input : xts       intersection points (m,2)
%         ids       intersection indices (m,2)
%         n         number of fractures
% output: gxs,gds   (n) cell
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [gxs,gds] = Group(xts,ids,n)
gds = cell(n,1);
gxs = cell(n,1);
for i=1:size(ids,1)
    I = ids(i,1);
    J = ids(i,2);
    gds{I} = [gds{I},J];
    gds{J} = [gds{J},I];
    gxs{I} = [gxs{I};xts(i,:)];
    gxs{J} = [gxs{J};xts(i,:)];
end
```

Group groups given intersection points based on their associated fracture indices.

2.6.12 FarthestPoints

```
% FarthestPoints
% finds two farthest points in a set of nD points
%
% Usage :
%    [p1,p2] = FarthestPoints(pts)
%
% input : pts       (n,2)
% output: p1,p2     two farthest points
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [p1,p2] = FarthestPoints(pts)
[~,idx] = max(pdist(pts,'euclidean'));
[I,J] = PDistIndices(size(pts,1));
p1 = pts(I(idx),:);
p2 = pts(J(idx),:);
```

FarthestPoints finds two farthest points from each other in a given set of n-dimensional points.

2.6.13 PDistIndices

```
% PDistIndices
% finds indices of results from `pdist` function
%
% Usage :
%    [I,J] = PDistIndices(n)
%
% input : n         number of points
% output: I,J       indices

% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function [I,J] = PDistIndices(n)
[I,J] = find(tril(ones(n),-1));
```

PDistIndices produces indices information for "*pdist*" function.

2.6.14 Occurrence

```
% Occurrence
% finds number of occurrence of a point in set of nD points
%
% Usage :
%    k = Occurrence(pt,pts,tol)
%
% input : pt        point
%         pts       points
%         tol       tolerance of distance
% output: k         occurrence number
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function k = Occurrence(pt,pts,tol)
if nargin<3; tol = 1e-9; end
[m,n] = size(pts);
k = true(m,1);
if tol~=0
    for i=1:n
        k = k & (abs(pts(:,i)-pt(i))<tol);        %relative match
    end
else
    for i=1:n
        k = k & (pts(:,i)==pt(i));                %absolute match
    end
end
k = sum(k);
```

Occurrence determines occurrence of any point in a set of n-dimensional points.

2.6.15 ConnectivityMatrix

```
% ConnectivityMatrix
```

```
% computes connectivity matrix of fracture network
%
% Usage :
%    cm = ConnectivityMatrix(ids,n,full,mat,fnm)
%
% input : ids       intersection indices
%         n         number of fractures
%         full      if true returns full matrix
%         mat       if false sparse form of results
%         fnm       fracture network
% output: cm        (n,n)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
function cm = ConnectivityMatrix(ids,n,full,mat,fnm)
if nargin==5                                        %to find indices if not provided
    if iscell(fnm)
        [~,ids,~] = PolysX3D(fnm);                  %3D fracture network
    else
        [~,ids,~] = LinesX2D(fnm);                  %2D fracture network
    end
    n = size(fnm,1);
end
if nargin<4; mat = true; end
if nargin<3; full = false; end
cm = zeros(n,n);
for i=1:size(ids,1)
    if iscell(ids)
        I = ids{i}(1);
        J = ids{i}(2);
    else
        I = ids(i,1);
        J = ids(i,2);
    end
    cm(I,J) = 1;
    if full; cm(J,I) = 1; end
end
if ~mat; cm = sparse(cm); end
```

ConnectivityMatrix generates connectivity matrix based on intersection indices for a fracture network. If fracture network was provided it applies intersection analysis to find intersection indices. The function accepts two- or three-dimensional fractures networks.

2.6.16 FullCM

```
% FullCM
% returns full form of connectivity matrix (cm)
%
% Usage :
%    fcm = FullCM(cm)
%
% input : cm        sparse/matrix of connectivity
% output: fcm       full matrix of cm
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
```

```
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function fcm = FullCM(cm)
if issparse(cm)
    fcm = full(cm);
else
    fcm = cm;
end
fcm = fcm+fcm';
```

FullCM builds full connectivity matrix based on sparse or triangular connectivity matrix.

2.6.17 FNMToGraph

```
% FNMToGraph
% creates Graph from fracture network
%
% Usage :
%    [G,cm] = FNMToGraph(ids,n,fnm)
%
% input : ids       intersection indices
%         n         number of fractures
%         fnm       fracture network 2D or 3D
% output: G         Graph
%         cm        connectivity matrix
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function [G,cm] = FNMToGraph(ids,n,fnm)
if nargin==3
    if iscell(fnm)
        [~,ids,~] = PolysX3D(fnm);                %3D fracture network
    else
        [~,ids,~] = LinesX2D(fnm);                %2D fracture network
    end
    if isempty(ids); G = empty; return; end
    n = size(fnm,1);
end
cm = ConnectivityMatrix(ids,n,false,false);
G = biograph(cm,num2str(linspace(1,n,n)'));       %Matlab graph structure
```

FNMToGraph generates graph structure based on intersection indices.

2.6.18 LoadColormap

```
% LoadColormap
% updates current colormap from file
%
% Usage :
%    LoadColormap(fname)
%
% input : fname     filename
```

```
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function LoadColormap(fname)
load(fname,'cmap');
set(gcf,'Colormap',cmap);
```

LoadColormap loads a given "*colormap*" file and applies it to current figure.

2.6.19 SaveColormap

```
% SaveColormap
% saves current colormap as file
%
% Usage :
%    SaveColormap(fname)
%
% input : fname     filename
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function SaveColormap(fname)
cmap = get(gcf,'Colormap');
save(fname,'cmap');
```

SaveColormap saves current "*colormap*" to a file.

2.6.20 SecondsToClock

```
% SecondsToClock
% converts seconds to clock format as string
%
% Usage :
%    clk = SecondsToClock(snd)
%
% input : snd       seconds
% output: clk       clock string
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function clk = SecondsToClock(snd)
h = floor(snd/3600);
m = floor((snd-(h*3600))/60);
s = rem(snd,3600)-m*60;
clk = sprintf('%02d:%02d:%05.2f',h,m,s);
```

SecondsToClock converts given seconds to "time (clock) format".

2.6.21 Colorise

```
% Colorise
% returns colors based on given data
%
% Usage :
%    colors = Colorise(x,cmap)
%
% input : x         (n)
%         cmap      colormap
% output: colors    (64,3)
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function colors = Colorise(x,cmap)
if nargin<2; cmap = colormap(jet); end
y = int32(Scale(x,1,64));
colors = cmap(y,:);
```

Colorise maps given data into a specified "*colormap*".

2.6.22 ShowFNM

```
% ShowFNM
% shows 2D or 3D fracture network
%
% Usage :
%    ShowFNM(fnm,La)
%
% input : fnm       (n,4) for 2D or cell for 3D
%         La        cluster labels
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function ShowFNM(fnm,La)
if nargin<2; La = []; end
if iscell(fnm)
    cla
    DrawPolys3D(fnm,La);                          %3D fracture networks
else
    cla
    DrawLines2D(fnm,La);                          %2D fracture networks
end
```

ShowFNM visualises given two- or three-dimensional fracture network.

2.6.23 Round

```
% Round
% rounds x to an arbitrary (dp) decimal
%
% Usage :
%    y = Round(x,dp)
```

```
%
% input : x         any
%         dp        decimal point, default=no decimal
% output: y         rounded output
%
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Nov 2013
function y = Round(x,dp)
if nargin<2; dp = 1.0; end
y = round(x/dp)*dp;
```

Round rounds data up to a given precision.

2.7 Example Full Programs

2.7.1 Example: Simulation of 2D Connectivity Index

By means of the provided functions Connectivity Index (CI) can be easily evaluated for two-dimensional fracture network model. The following full program code demonstrates the required stages and setup. Figure 2.1 shows the resulting maps.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

%% Simulation for Connectivity Index (CI)
n = 500;
simN = 30;
gm = 25;                                        %grid dimension vertically
gn = 25;                                        %grid dimension horizontally
cm = int32(floor(gm/2)+1);
cn = int32(floor(gn/2)+1);
CI = zeros(gm,gn);
tic
kappa = 10;
for i = 1:simN
    lines = GenFNM2D(n,3*pi/4,kappa,0.05,0.5);
    La = LinesToClusters2D(lines);
    ci = ConnectivityIndex2D(lines,La,gm,gn,cm,cn);
    CI = CI+ci;
    fprintf(1,'Real#:%04d, Total Elapsed Time:<%s>\n',i,SecondsToClock(toc));
end
CI = CI/simN;

%% Visualisation
clf
subplot(131);
```

```
[X,Y] = LinesToXYnan2D(lines);
plot(X,Y,'k-')
Titles2D('-=[')

colormap(jet);
subplot(132);
w = 0.5/gn;
h = 0.5/gm;
imagesc([w,1-w],[h,1-h],CI);
set(gca,'YDir','normal');
Titles2D('-=[');

subplot(133);
sCI = Smooth(CI);
contourf(linspace(0,1,25),linspace(0,1,25),sCI,20);
shading flat
Titles2D('-=[');

print('-dpng','-r600','CI2D Example');                %to export result as image
```

As can be seen in Fig. 2.1 the simulated fracture network is anisotropic towards North-West, South-East. This is due to the setting in the code i.e., main orientation $3 \times \frac{pi}{4}$ and κ equal to 10. The resulting CI maps show the anisotropy.

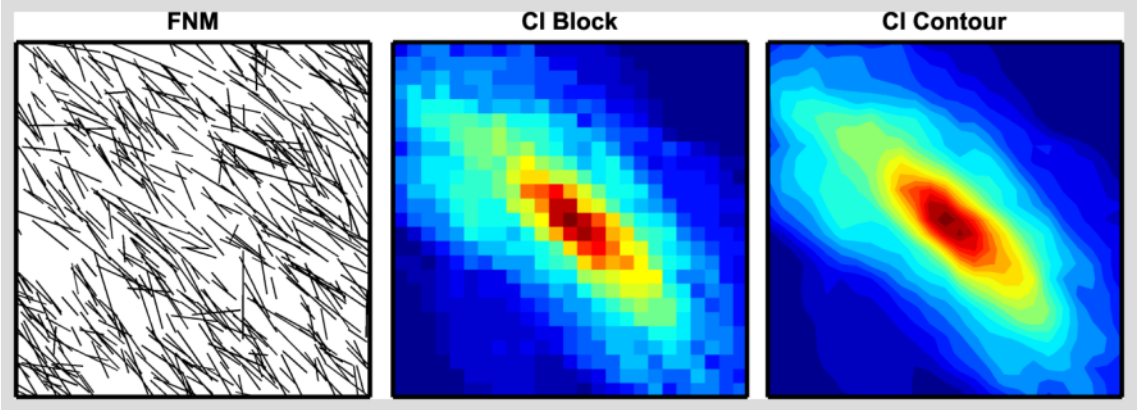


Figure 2.1: Results of evaluating CI on anisotropic fracture network.

2.7.2 Example: Two-dimensional Line Sampling

Line sampling is common stage for evaluating connectivity measures including CI and CF. The following full program code shows how generating line samples can be conducted by means of provided functions in the AFNM package.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013
```

```
clear all
clc

lines = RandLinesInPoly2D(1000,0.09,0.01,0,pi/2);      %generating line samples
[X,Y] = LinesToXYnan2D(lines);
clf
plot(X,Y,'k-')
Titles2D()
```

Note that any function provides quick help on its parameters and usage upon request by right-click in Matlab environment as shown for **`RandLinesInPoly2D`** in Fig. 2.2.

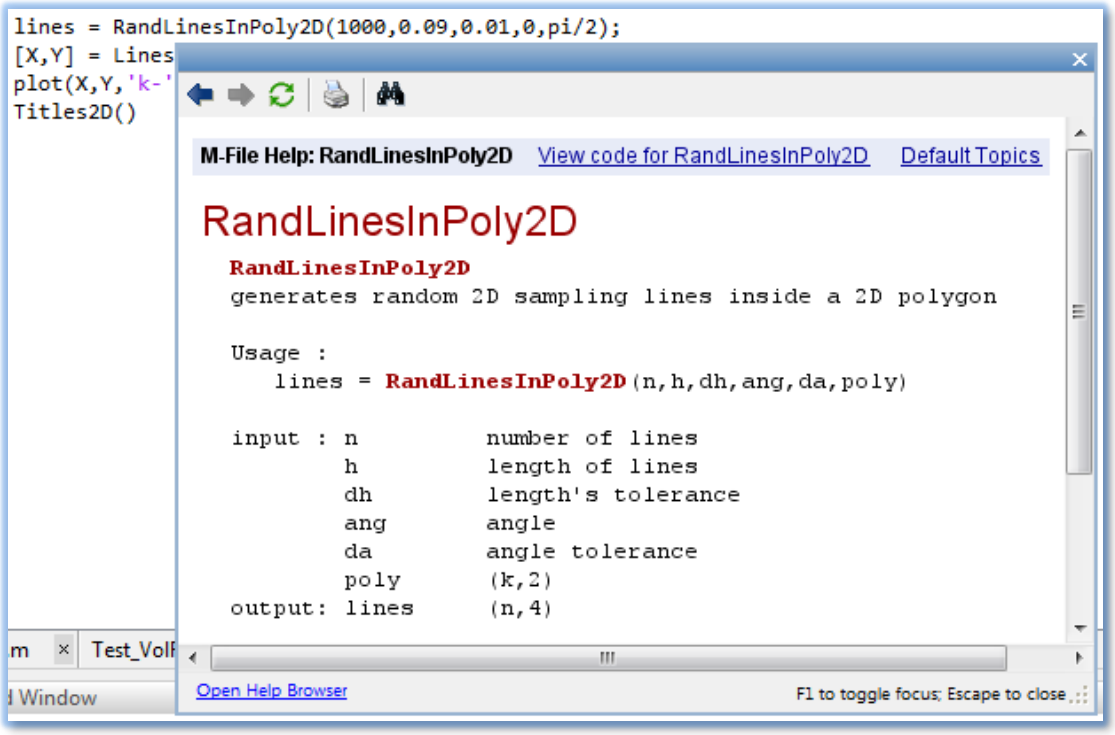


Figure 2.2: Quick help is available any time for all the functions.

Variation in the function **`RandLinesInPoly2D`** parameters results in various setting of line samples as the examples shown in Fig. 2.3.

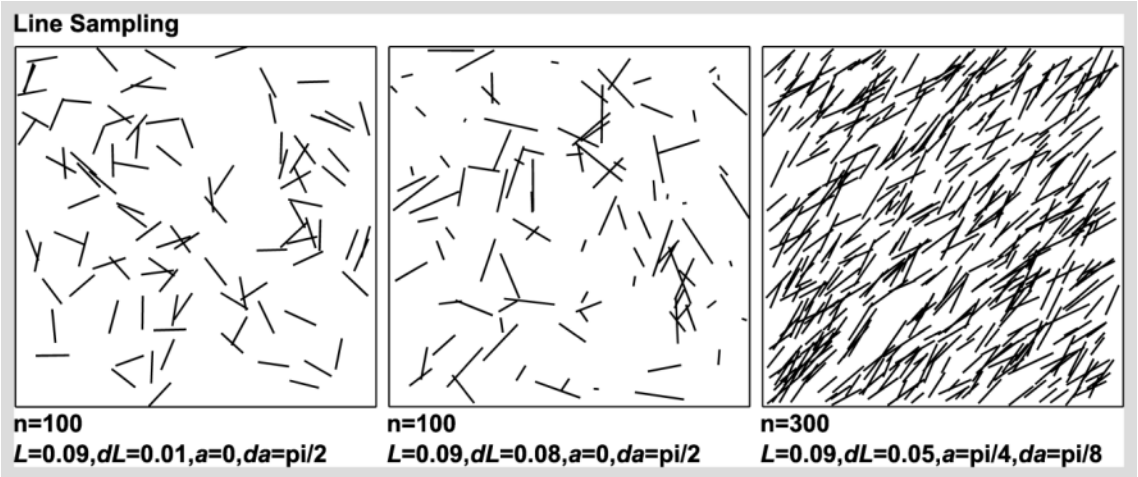


Figure 2.3: Various setting of line samples can be used in the determination of directional connectivity measures, for example.

2.7.3 Example: Simulation of 3D Connectivity Index

The evaluation of three-dimensional CI is simple and straightforward by means of the provided functions. The following full program code evaluates CI on a simulation of 30 realisations from a three-dimensional fracture network model. Each realisation includes 70 three-dimensional polygonal fractures which are randomly located and oriented in a unit cube.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

%% CI3D
w = 0.2; h = 0.2; d = 0.2;
m = 1/w; n = 1/h; o = 1/d;
i = floor(m/2)+1; j = floor(n/2)+1; k = floor(o/2)+1;
sup1 = Sup3D([i/m-w/2,j/n-h/2,k/o-d/2],[w,h,d]);
CI = zeros(m,n,o);
tic
for s=1:30                                              %simulation number
    plys = GenFNM3D(70,deg2rad(15),deg2rad(15),0);      %anisotropic
    [~,~,La] = PolysX3D(plys);
    for i=1:m
        for j=1:n
            for k=1:o
                pt = [i/m-w/2,j/n-h/2,k/o-d/2];
                sup2 = Sup3D(pt,[w,h,d]);
                CI(i,j,k) = CI(i,j,k)+SupCSup3D(sup1,sup2,plys,La);
```

```
            end
        end
    end
    fprintf(1,'Real.#: %d, Total Elapsed Time:<%s>\n',s,SecondsToClock(toc));
end

clf
DrawPolys3D(plys);
```

The resulting CI matrix which is three-dimensional volumetric data can be visualised by means of **`VolRender3D`** and **`DrawSlices3D`** (see also Fig. 2.4).

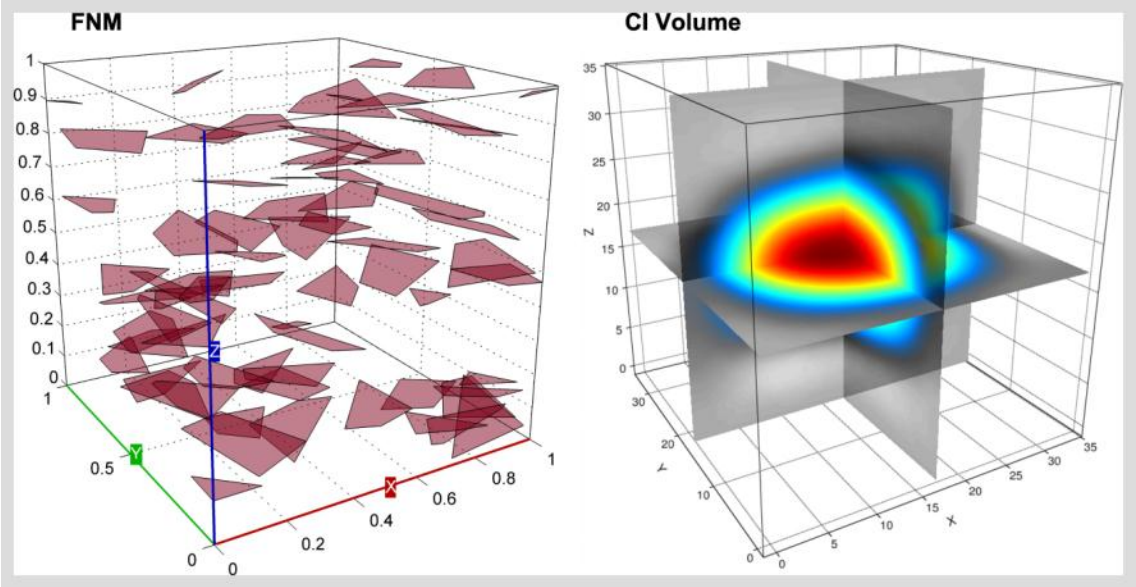


Figure 2.4: An example of CI3D on anisotropic fracture network.

2.7.4 Example: Intersection Analysis and Fracture Clusters

Conducting the intersection analysis and assessing fracture clusters are easy tasks by means of the provided functions in the AFNM as shown in the following full program code.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

%% Fracture Network models
fnm2 = GenFNM2D(150,0,0,0.01,0.7);
```

```
fnm3 = GenFNM3D(150,pi/3,pi/3,0,0.25);

%% Intersection Analysis >> Clusters
[xts2,ids2,La2] = LinesX2D(fnm2);
[xts3,ids3,La3] = PolysX3D(fnm3);

%% Visualisations
clf
subplot(121);
ShowFNM(fnm2,La2);                  % a generic visualisation function which handles
                                    %    automatically 2D and 3D fractures and
subplot(122);                       %    and associated cluster data
ShowFNM(fnm3,La3);
```

The numeric results for the two-dimensional fracture network are as follows.

```
| xts1 =              | ids1 =          | La1 =    |
|   0.4720    0.7168  |     2      18   |      -1  |
|   0.2915    0.5062  |     2      61   |       1  |
|   0.4091    0.6434  |     2      73   |       1  |
|      :         :    |     :       :   |       :  |
```

For the three-dimensional fracture network the results are as follows.

```
| xts2 =              | ids2 =          | La2 =    |
|    [1x3 double]     |    [1x2 int32]  |      -1  |
|    [1x3 double]     |    [1x2 int32]  |       1  |
|    [2x3 double]     |    [1x2 int32]  |       2  |       %two intersection points
|          :          |         :       |       :  |
```

Fracture clusters are visualised by the help of clusters labels (*La*) passed to function **`ShowFNM`** (Fig. 2.5).

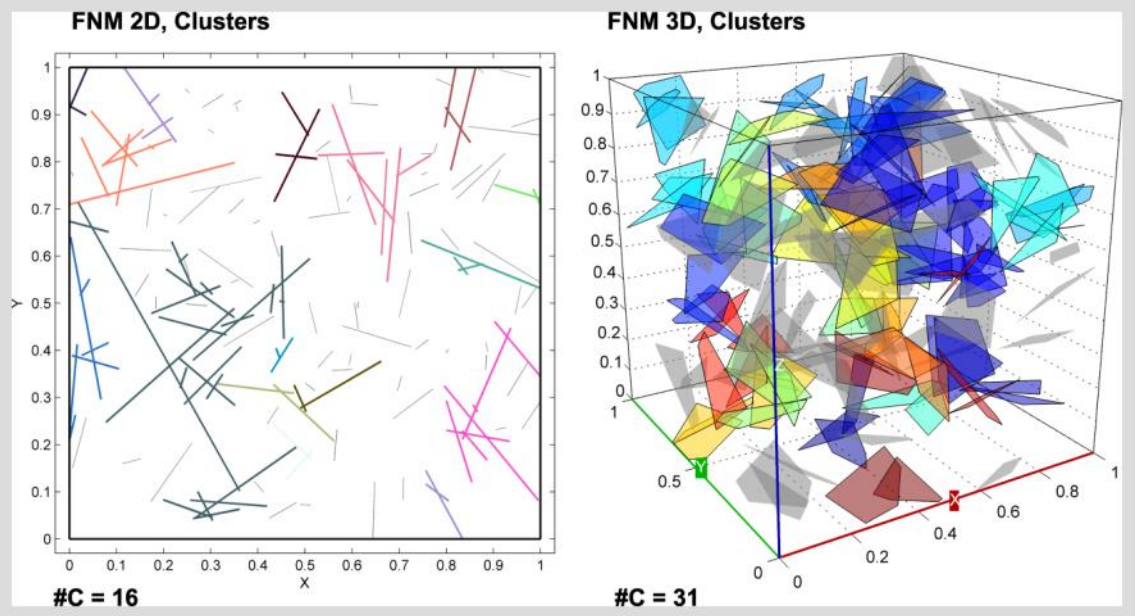


Figure 2.5: Fracture clustering is conducted on two- and three-dimensional fracture networks.

2.7.5 Example: Density Analysis

The density of fracture network can be found via different methods including density of fracture centroids (DFC, FCD). Here however the following program code demonstrates better solution which is based on cell sampling, i.e., fracture density (Fn). The function **`Density2D`** conducts the evaluation and results in density matrix which can be visualised as block or contour maps.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

%%
n = 200;
gn = 20;
sf = gn;
lines = GenFNM2D(n,0,0,0.05,0.5);
[dn,x,y] = Density2D(lines,gn,gn);

sdn = Smooth(dn,1);
[X,Y] = LinesToXYnan2D(lines);

clf
subplot(121);
imagesc(x,y,dn);
set(gca,'YDir','normal');
hold on
plot(X,Y,'k-','LineWidth',0.7)
Titles2D('-=[')

subplot(122);
contourf(0:1/(sf-1):1,linspace(0,1,sf),sdn,30);
shading flat
hold on
plot(X,Y,'k-','LineWidth',0.7)
Titles2D('-=[')
```

The output of the above program code is shown in Fig. 2.6. The same concept can be used for three-dimensional fracture networks.

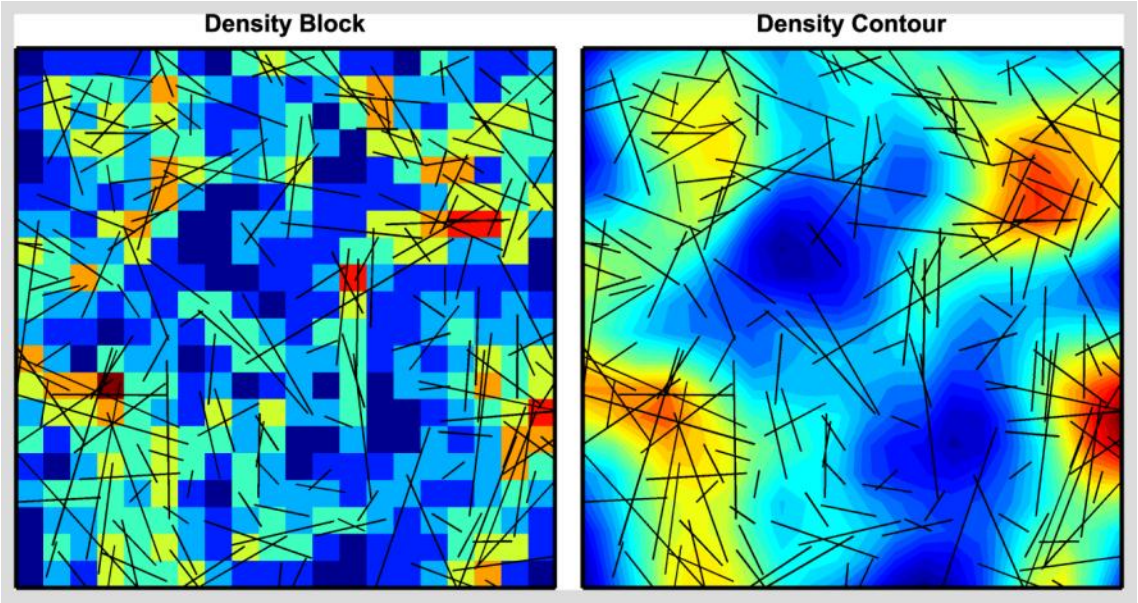


Figure 2.6: An example of density of fracture network; block and contour maps.

2.7.6 Example: Backbone Extraction

Backbone of two-dimensional fracture networks can be efficiently extracted by means of the function **Backbone2D**.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

fnm2 = GenFNM2D(150);                           %fracture network

clf
subplot(121);
DrawLines2D(fnm2,0,0,'-=[');                    %visualisation of fnm

bbn = Backbone2D(fnm2,true);                    %backbone extraction

if ~isempty(bbn)
    subplot(122);
    DrawLines2D(bbn,0,0,'-=[');                 %visualisation of the backbone
end
```

The resulting backbone from the above code is shown in Fig. 2.7.

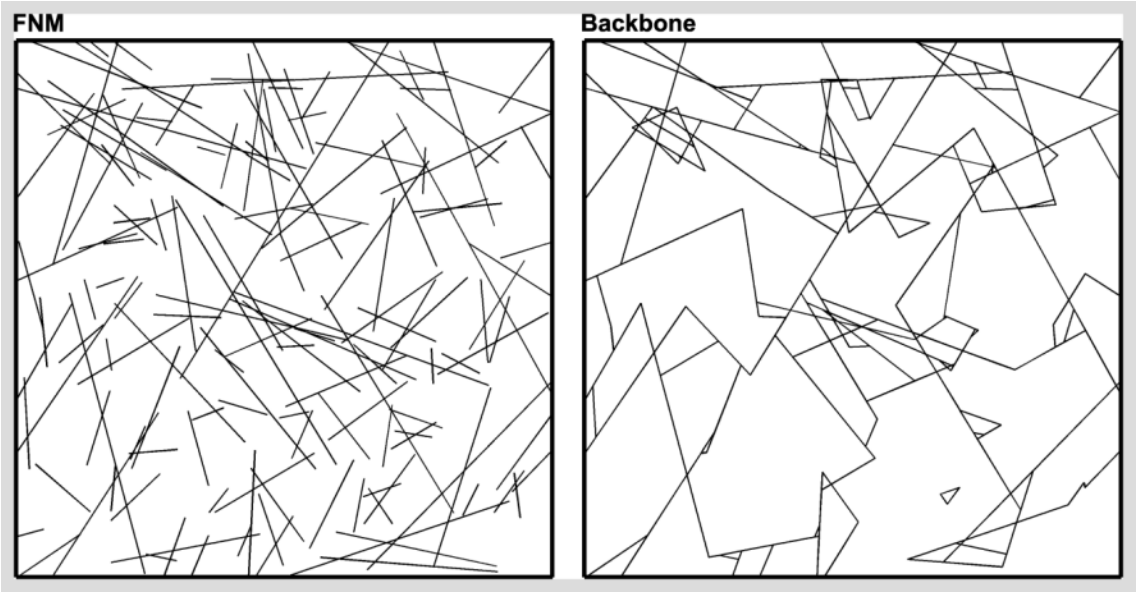


Figure 2.7: An example of backbone of fracture network.

The **BackboneToNodesEdges2D** function was used in the following program code to determine the popularity (centrality) of each node in the *graph* network.

```
% Part of package: Alghalandis Fracture Network Modelling (AFNM)
% Author: Younes Fadakar Alghalandis
% email: younes.fadakar@yahoo.com
% Copyright (c) 2011-2012-2013-2014 Younes Fadakar Alghalandis
% All rights reserved.
% Updated: Oct 2013

clear all
clc

lines = GenFNM2D(300);                          %2D fracture network
bbn = Backbone2D(lines,true,[],0);              %backbone
[nodes,edges] = BackboneToNodesEdges2D(bbn);    %nodes and edges: graph structure
nn = cellfun(@numel,nodes.values);              %popularity of nodes
pts = Stack(nodes.keys);                        %node locations

clf
subplot(121);
DrawLines2D(lines,0,[0,0,0],'-=[');

subplot(122);
DrawLines2D(bbn,0,[0,0,0],'-=[');
hold on
scatter(pts(:,1),pts(:,2),10,nn,'filled');      %colourised by popularity
```

In Fig. 2.8(*right*) blue, green and red dots correspond to 2, 3 and 4 connected edges respectively.

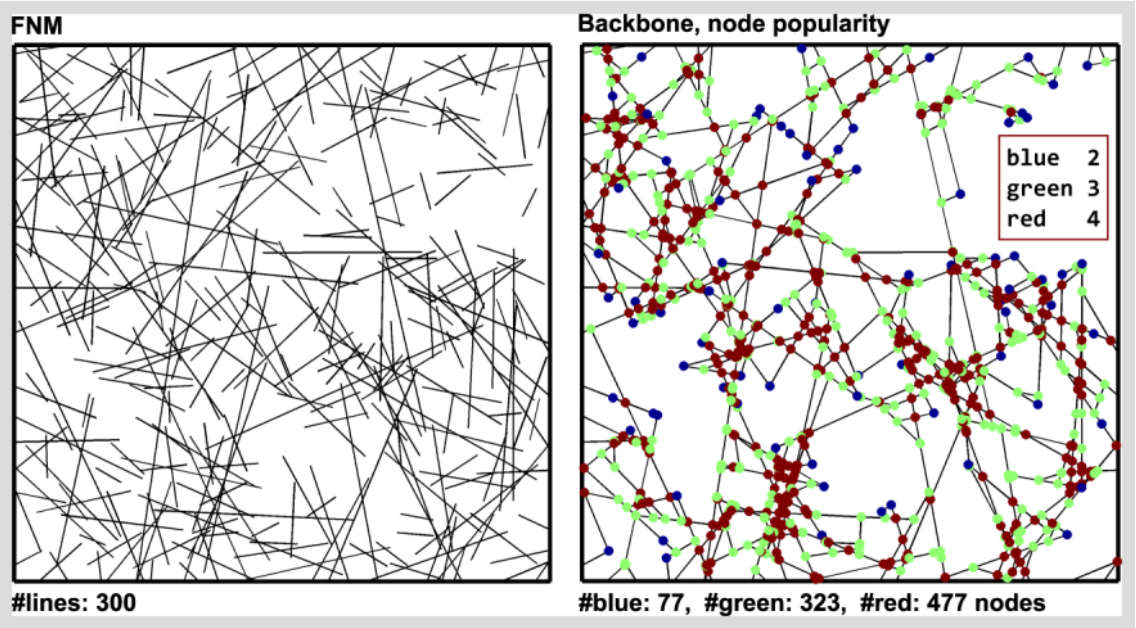


Figure 2.8: An example of backbone of fracture network, node centrality evaluation.

Index

References

Alemanni A, Battaglia M, Bigi S, Borisova E, Campana A, Loizzo M, Lombardi S (2011) A three dimensional representation of the fracture network of CO_2 reservoir analogue (Latera caldera, central Italy). J Energy Procedia 4:3582-3587.

Allard D, HERESIM Group (1993) On the connectivity of two random set models: the truncated Gaussian and the Boolean. in: Soares A (ed.) Geostatistics Troia'92, Kluwer Academic Publishers, p467-478.

Andersson J, Dverstorp N (1987) Conditional Simulations of Fluid Flow in Three-Dimensional Networks of Discrete Fractures. J Water Resource Research, 23(10):1876-1886.

Andersson J, Shapiro AM, Bear J (1984) A stochastic model of a fractured rock conditioned by measured information. J Water Resources Research, 20(1):79-88.

Audigane P, Royer JJ, Kaieda H (2002) Permeability characterization of the Soultz and Ogachi large-scale reservoir using induced micro-seismicity, Geophysics, 67(1):204-211.

Back PE (2001) Sampling strategies and data worth analysis for contaminated land. Statens Geotekniska Institut, Varia 500, pp98.

Baddeley A (2010) Analysing spatial point patterns in R. Workshop Notes http://www.csiro.au/resources/Spatial-Point-Patterns-in-R, CSIRO, Australia, pp232.

Baddeley A, Gregori P, Mateu J, Stocia R, Stoyan D (2006) Case Studies in Spatial Point Process Modeling. Springer, pp306.

Baecher GB (1983) Statistical Analysis of Rock Mass Fracturing. J Mathematical Geology, 15(2):330-348.

Bairos KP (2012) Insights from use of a 3D Discrete Fracture Network Numerical Model for Hydraulic Test Analysis. MSc Thesis, The University of Guelph, Ontario, Canada.

Baisch S, Weidler R, Voros R, Wyborn D, de-Graaf L (2006) Induced seismicity during the stimulation of a geothermal HFR reservoir in the Cooper Basin, Australia. Bull Seismol Soc Am 96(6):2242–2256

Balberg I (1986) Connectivity and conductivity in 2D and 3D fracture systems. in: Englman R, Jaeger Z (eds.) Proceedings of the intl. Conference on Fragmentation, Form and Flow in Fractured Media, England, Ann. Isr. Phys. Soc., 8:89-101.

Berkowitz B (1995) Analysis of fracture network connectivity using percolation theory. J Mathematical Geology, 27(4):467-483.

Berkowitz B (2002) Characterizing flow and transport in fractured geological media: A review. J Advances in Water Resources, 25:861-884.

Billaux D, Chilès JP, Hestir K, Long J (1989) Three-Dimensional Statistical Modelling of a Fractured Rock Mass-an Example from the Fanay-Augeres Mine. intl. J Rock Mechanics and Mining Science, 26(3,4):281-299.

Blocher MG, Cacace M, Lewerenz B, Zimmermann G (2010) Three dimensional modelling of fractured and faulted reservoirs: Framework and implementation. J Chemie der Erde, 70(S3):145-153.

Board M (1989) UDEC (Universal Distinct Element Code), User's Manual. Itasca Consulting Group, pp431.

Boissonnat JD, Yvinec M (1998) Algorithmic Geometry. Cambridge University Press, pp519.

Bour O, Davy P (1997) Connectivity of random fault networks following a power law fault length distribution. J Water Resources Research, 33(7):1567-1583.

Bour O, Davy P (1999) Clustering and size distributions of fault patterns: theory and measurements. J Geophysical Research Letters, 26(13):2001-2004.

Bruel D (2007) Using the migration of the induced seismicity as a constraint for fractured Hot Dry Rock reservoir modeling. Int J Rock Mechanics, Mining Sciences, 44(8):1106-1117.

Cacas MC, Daniel JM, Letouzey J (2001) Nested geological modelling of naturally fractured reservoirs. J Petroleum Geoscience, 7:43-52.

CFCFF (1996) Rock Fractures and Fluid Flow: Contemporary Understanding and Applications. National Academy Press, pp568.

Chernoutsan AI, Egorov AV, Manzhirov AV, Polyanin AD, Polyanin VD, Popov VA, Putyatin VB, Repina YV, Safrai VM, Zhurov AI (2011) A Concise Handbook Of Mathematics, Physics, And Engineering Sciences. Polyanin A.D, Chernoutsan A.I (eds.), CRC Press, pp1080.

Cherpeau N, Caumon G, Caers J, Levy B (2012) Method for Stochastic Inverse Modeling of Fault Geometry and Connectivity Using Flow Data, J Mathematical Geosciences 44:147-168.

Chilès JP (1988) Fractal and Geostatistical Methods for Modeling of a Fracture Network. J Mathematical Geology, 20(6):631-654.

Chilès JP (2004) Stochastic Modeling of Natural Fractured Media: A Review. in: Leuangthong O, Deutsch C.V (eds.) Geostatistical Banff, p285-294.

Chilès JP, Delfiner P (1999) Geostatistics: modeling spatial uncertainty. Wiley, New York, p695.

Chilès JP, de-Marsily G (1993) Stochastic models of fracture systems and their use in flow and transport modelling. in: Bear J, Tsang C.F, de Marsily G (eds.) Flow And Contaminant Transport in Fractured Rock, Academic Press, p169-236.

Corrochano EB (2005) Handbook of Geometric Computing, Applications in Pattern Recognition, Computer Vision, Neural Computing, and Robotics. Springer, pp774.

Cosgrove JW (1998) The role of structural geology in reservoir characterisation. in: Coward M.P, Daltaban T.S, Johnson H (eds.) Structural Geology in Reservoir Characterisation, Geological Society, Special Publication, p127:1-13.

Cressie N (1991) Statistics for Spatial Data. John Wiley & Sons, pp900.

de-Dreuzy JR, Erhel J (2003) Efficient algorithms for the determination of the connected fracture network and the solution to the steady-state flow equation in fracture networks. J Computer & Geosciences, 29:107-111.

Dershowitz W (1992) Interpretation of fracture spacing and intensity. in: Tillerson, Wawersik (eds.) Rock Mechanics, Balkema, Rotterdam, ISBN 9054100451.

Dershowitz W, Busse R (1996) A stochastic approach for fracture set definition. in: Aubertin M, Hassani F, Mitri H (eds.) Rock Mechanics: Tools and Techniques, 2:1809-1813.

Dershowitz W, Carvalho J (1996) Key-block tunnel stability analysis using realistic fracture patterns. in: Aubertin M, Hassani F, Mitri H (eds.) Rock Mechanics: Tools and Techniques, 2:1747-1751.

Dershowitz WS (1984) Rock joint systems. PhD Thesis, MIT, Cambridge, MA.

Dershowitz WS, Einstein HH (1988) Characterizing rock joint geometry with joint system models. J Rock Mechanics and Rock Engineering, 21:21-51.

Dershowitz WS, La-Pointe PR, Doe TW (2000) Advances in Discrete Fracture Network Modeling. in: proceeding of the US EPA/NGWA Fractured Rock Conference, p882-894.

Dershowitz WS, Lee G, Geier J, Foxford T, La-Pointe P, Thomas A (1998) FracMan Version2.6 Interactive Discrete Feature data Analysis, Geometric Modeling and Exploration Simulation, User Documentation, Report 923-1089, Golder Associates Inc, Seattle, Washington.

Descombes, X, Zerubia J (2002) Marked Point Process in Image Analysis. IEEE Signal Processing Magazine, p77-84.

Deutsch CV, Cockerham PW (1994) Practical considerations in the application of simulated annealing to stochastic simulation. Mathematical Geology, 26:67-82.

Deutsch CV, Journel AG (1998) GSLIB, Geostatistical Software Library and User's Guide. Applied Geostatistics, Oxford University Press, pp384.

Diggle P (1983) Statistical analysis of spatial point patterns. Academic Press, pp148.

Diggle P (2003) Statistical Analysis of Spatial Point Patterns. Edward Arnold Publication, pp159.

Dimov I, Dimova S, Kolkovska N (2011) Numerical Methods and Applications. Springer, pp512.

Diodato DM (1994) A compendium of fracture flow models. Energy Systems Division, Center for Environmental Restoration Systems, Argonne National Lab, pp94.

Dowd PA, Martin JA, Xu C, Fowell RJ, Mardia KV (2009) A three-dimensional fracture network data set for a block of granite. Intl. J Rock Mechanics & Mining Sciences, 46:811-818.

Dowd PA, Xu C, Mardia KV, Fowell RJ (2007) A comparison of methods for the stochastic simulation of rock fractures. J Mathematical Geology, 39:697-714.

Duda RD, Hart PE (1972) Use of the Hough Transform to detect lines and curves in pictures. Comm. ACM 15:11-15.

Duong T (2004) Bandwidth selectors for multivariate kernel density estimation. PhD Dissertation, School of Mathematics and Statistics, University of Western Australia, pp173.

Dverstorp B (1991) Analyzing flow and transport in fractured rock using the discrete fracture network concept. Royal Institute of Technology, Stockholm, BN:Trita-VBI-151, pp64.

Einstein HH (1993) Modern developments in Discontinuity Analysis-The persistence-Connectivity problem. In Comprehensive Rock Engineering, Rock testing and site characterization, 3:193–213.

Eisner L, Williams-Stroud S, Hill A, Duncan P, Thornton M (2010) Beyond the dots in the box: microseismicity-constrained fracture models for reservoir simulation. Lead Edge 29(3):326–333. doi:10.1190/1.3353730

Elfouly A (2000) Faults and Fractures Intersections Delineation as a Tool for Groundwater Detection Using Remote Sensing and Ground Penetrating Radar Techniques at Saint Catherine area, Southern Sinai, Egypt. ICEHM2000, Cairo University, Egypt, p293-310.

Fadakar-A Y, Dowd PA, Xu C (2012) Application of Connectivity Measures in Enhanced Geothermal Systems. in: proceedings of Australian Geothermal Energy Conference AGEC2012, Sydney, Australia, p62-66.

Fadakar-A Y, Dowd PA, Xu C (2013a) The RANSAC method for generating fracture networks from micro-seismic event data. J Mathematical Geosciences, 45:207-224

Fadakar-A Y, Dowd PA, Xu C (2013b) Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs. Proceedings, Stanford Geothermal Workshop SGW2013, Stanford, USA.

Fadakar-A Y, Dowd PA, Xu C (2013c) A Connectivity-Graph Approach to Optimising Well Locations in Geothermal Reservoirs. in: proceedings of Australian Geothermal Energy Conference AGEC2013, Brisbane, Australia, p111-115.

Fadakar-A Y, Dowd PA, Xu C (2014) Connectivity Field: A measure for characterising fracture networks, J Mathematical Geosciences (in press).

Fadakar-A Y, Xu C, Dowd PA (2011) A general framework for fracture intersection analysis: algorithms and practical applications. in: proceedings of Australian Geothermal Energy Conference AGEC2011, Melbourne, Australia, p15-20.

Fadakar-A Y, Xu C, Dowd PA (2013b) Connectivity Index and Connectivity Field towards fluid flow in fracture-based geothermal reservoirs. in proceedings of 38 Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, p417-427.

Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381-395.

Fouche O, Diebolt J (2004) Describing the geometry of 3d fracture systems by correcting for linear sampling bias. J Mathematical Geology, 36(1):33-63.

Freeden W, Zuhair-N M, Sonar T (2010) Handbook of Geomathematics. Springer, pp1338.

Freeze RA (1975) A stochastic-conceptual analysis of one-dimensional groundwater flow in non-uniform homogeneous media. J Water Resources Research, 11(5):725-741.

Gartner B, Schonherr S (1998) Smallest enclosing ellipses: An exact and generic implementations in C++. Informatik Series B, pp150.

Gavrilova ML (2008) Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence. Springer, pp312.

Geier JE, Lee K, Dershowitz WS (1988) Field Validation of Conceptual Models for Fracture Geometry. American Geophysical Union, 1988 Fall Meeting, San Francisco, EOS, Transactions, American Geophysical Union, 69(44):1177.

Ghaffari HO, Nasseri MHB, Young RP (2012) Fluid flow complexity in fracture networks: analysis with graph theory and LBM. Cornell University Library, http://arxiv.org/abs/1107.4918, pp9.

Goodman JE, O'Rourke J (2004) Handbook of Discrete and Computational Geometry. Chapman & Hall/CRC, pp1558.

Goodman RE, Shi G (1985) Block Theory and its Applications to Rock Engineering. Prentice-Hall intl., London, pp338.

Grasby SE, Allen DM, 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 Canada. Geological Survey of Canada. Open File 6914, pp322.

Grenon M, Hadjigeorgiou J (2003) Open Stope Stability Using 3D Joint Networks. J Rock Mechanics and Rock Engineering, 36(3):183-208.

Gringarten E (1997) Geometric Modeling of Fracture Networks, PhD Thesis, Stanford University, USA, pp128.

Gross JL, Yellen I (2004) Handbook of Graph Theory. CRC Press LLC, pp839.

Gupta AK, Adler PM (2006) Stereological Analysis of Fracture Networks along Cylindrical Galleries. J Mathematical Geology, 38(3):233-267.

Haining R (2004) Spatial Data Analysis, Theory and Practice. Cambridge University Press, UK, pp454.

Hanano M (2004) Contribution of fractures to formation and production of geothermal resources. Renewable and Sustainable Energy Reviews, 8:223-236.

Haneberg WC, Mozley PS, Moore JC, Goodwin LB (1999) Faults and Subsurface Fluid Flow in the Shallow Crust. American Geophysical Union.

Harris S, Ross J (2005) Beginning Algorithms. Wrox, 600p.

Harrison JP, Hudson JA (2000) Engineering Rock Mechanics. Pergamon, pp530.

Hartley RT, Zisserman A (2004) Multiple View Geometry in Computer Vision. Cambridge University Press, 672p.

Hayashi K, Willis-Richards J, Hopkirk RJ, Niibori Y (1999) Numerical models of HDR geothermal reservoirs-a review of current thinking and progress. J Geothermics 28:507-518.

Hernqvist L (2009) Characterization of the fracture system in Hard Rock for Tunnel Grouting. Licentiate Thesis, Chalmers University of Technology, Sweden.

Hestir K, Long JC (1990) Analytical expressions for the permeability of random two-dimensional Poisson fracture networks based on regular lattice percolation and equivalent media theories. J Geophysical Research 95 (B13):21565–21581.

Hofer M, Odehnal B, Pottmann H, Steiner T, Wallner J (2005) 3D Shape recognition and reconstruction based on line element geometry. In 10th IEEE Int Conference on Computer Vision, 2:1532–1538.

Hoshen J, Kopelman R (1976) Percolation and cluster distribution, I: Cluster multiple labeling technique and critical concentration algorithm. J Physics Review B14(8):3438-3445.

Hough PVC (1962) A method and means for recognizing complex patterns. U.S Patent 3,069,654.

Hsu T (2008) Writing Proofs. San Jose State University, pp44.

Huseby O, Thovert JF, Adler PM (1997) Geometry and topology of fracture systems. J Phys. A: Math. Gen. 30:1415-1444.

Illian J, Penttinen A, Stoyan H, Stoyan D (2008) Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley & Sons, England, pp557.

Jimenez-Rodriguez R, Sitar N (2008) Influence of Stochastic Discontinuity Network Parameters on the Formation of Removable Blocks in Rock Slopes. Rock Mechanics and Rock Engineering, 41(4):563-585.

Jimenez-Rodriguez R, Siter N (2006) Inference of discontinuity trace length distributions using statistical graphical models. Intl. J Rock Mechanics and Mining Sciences, 43:877-893.

Jing L (2003) A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering. Intl. J Rock Mechanics and Mining Sciences 40:283–353.

Jing L, Stephansson O (2007a) Fundamentals of Discrete Element Methods for Rock Engineering Theory and Applications. Elsevier, pp545.

Jing L, Stephansson O (2007b) The basics of fracture system characterization-field mapping and stochastic simulations. J Developments in Geotechnical Engineering, 85:147-177.

Kacewicz M (1994) Model-free estimation of fracture apertures with neural networks. J Mathematical Geology, 26(8):985–994.

Karvounis D, Jenny P (2011) Modeling of flow and transport in enhanced geothermal systems. in: Proceedings of 36th Workshop on Geothermal

Reservoir Engineering, Stanford University, Stanford, California, SGP-TR-191, pp8.

Kendall WS (2003) Models and simulation techniques from stochastic geometry, A course of Madison Probability Intern Program. The University of Warwick, pp163.

Kimmel R (2003) Numerical geometry of images: theory, algorithms and applications. Springer, pp209.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Koike K, Ichikawa Y (2006) Spatial correlation structures of fracture systems for deriving a scaling law and modeling fracture distributions. J Computer & Geosciences, 32:1079-1095.

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 Computer and Geotechnics 36:291-303.

Kulatilake PHSW, Wathugala DN, Stephansson O (1993) Joint network modelling with a validation exercise in Stripa Mine, Sweden. Int J Rock Mech Min Sci 30:503–526

Kvartsberg S (2010) Hydrogeological Characterisation of A Fracture Network. MSc Thesis, Chalmers University of Technology, Sweden, pp95.

Lacazette (2011) Natural Fracture Types. http://naturalfractures.com/1.1.1.htm, accessed July 2011.

Lantuejoul C (2002) Geostatistical simulation, models and algorithms. Springer, Berlin

La-Pointe PR, Hudson JA (1985) Characterization and interpretation of rock mass joint patterns. Geological Society of America, Special paper 199, Denver, p37.

La-Pointe PR, Wallmann PC (1996) Three-dimensional flow analysis in complex fracture networks through graph theory-based search algorithms. in: Aubertin M, Hassani F, Mitri H (eds.) Rock Mechanics: Tools and Techniques, 2:1787-1792.

Laslett GM (1982) Censoring and edge effects in areal and line transect sampling of rock joint traces. J Mathematical Geology, 14(2):125-140.

Lee C, Lee C, Yeh H, Lin H (2010) Modeling spatial fracture intensity as control on flow in fractured rock. Environmental Earth Sciences, 63(6):p1199-1211.

Lee CH (1990) Flow in Fractured Rock. PhD Thesis, The University of Arizona, USA.

Lee JS, Veneziano D, Einstein HH (1990) Hierarchical fracture trace model. In: Hustrulid W, Johnson GA (eds) Rock contributions and challenges, proceeding. 31st US Rock Mechanics Symposium, Balkema, Rotterdam, pp 261–268

Leeds (2011) Stochastic Rock Fractures. http://www.leeds.ac.uk/stochasticrockfractures/ accessed Jan 2011.

Li Y, Anderson-Sprecher R (2011) Detection of Fluid Signals in the 1985 Yellowstone Earthquake Swarm, J Mathematical Geosciences, 43:293-304.

Liu GR, Quek SS (2003) Finite Element Method, A Practical Course. Elsevier Science, pp361.

Locsin JLZ, Einstein HH (2005) Modeling Rock Fracture Intersections-Application in the Boston Area and Estimation of the Well-Test Flow Dimension. Technical Report in Fracture Flow Research, Massachusetts Institute Of Technology, pp341.

Long JCS, Remer JS, Wilson CR, Witherspoon PA (1982) Porous Media Equivalents for Networks of Discontinuous Fractures. J Water Resources Research, 18(3):645-658.

Manzocchi T, Ringrose PS, Underhill JR (1998) Flow through fault systems in high-porosity sandstones. in: Coward M.P, Daltaban T.S, Johnson H (eds.) Structural Geology in Reservoir Characterisation, Geological Society, London, Special Publications, 127:65-82.

Mardia KV, Nyirongo VB, Walder AN, Xu C, Dowd PA, Fowell RJ, Kent JT (2007a) Markov Chain Monte Carlo Implementation of Rock Fracture Modelling. J Mathematical Geology, 39:355-381.

Mardia KV, Walder AN, Xu C, Dowd PA, Fowell RJ, Nyirongo VB, Kent JT (2007b) A line finding assignment problem and rock fracture modelling. In: Upadhaya SK, Singh U, Dey DK (eds) Bayesian statistics and its applications. Anamaya Pubs, New Delhi, pp 319–330

Matas J, Chum O (2004) Randomized RANSAC with Td,d test. Image and Vision Computing, 22(10): 837-842.

Matyka M, Khalili A, Koza Z (2008) Tortuosity-porosity relation in the porous media flow. http://arxiv.org/abs/0801.3316v1, pp8.

Mauldon M (1992) Relative probabilities of joint intersections. In: Tillerson, J. R., Wawersik, W. R. (eds.), Rock mechanics; Proc., 33rd U.S. Symposium, A.A. Balkema, location varies, pp. 767-774.

Mauldon M (1994) Intersection probabilities of impersistent joints. Intl.J Rock Mechanics & Mining Sciences. Geomech. Abst. 31(2):107-115.

McClure M (2012) Modeling and Characterization of Hydraulic Stimulation and Induced Seismicity in Geothermal and Shale Gas Reservoirs. Stanford Geothermal Program, SGP-TR-199, pp369.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21(6):1087

Meyer T, Einstein HH (2002) Geologic stochastic modeling and connectivity assessment of fracture systems in the Boston area. Rock Mech Rock Eng 35(1):23–44

Michaelides K, Chappell A (2009) Connectivity as a concept for characterising hydrological behaviour. J Hydrological Processes, DOI: 10.1002/hyp.7214, 23:517-522.

MIT (2010) The Future of Geothermal Energy. Technical Report, U.S. Government http://mitei.mit.edu/publications/reports-studies/future-geothermal-energy, pp372.

Mohais R, Xu C, Dowd PA (2011a) An analytical model of coupled fluid flow and heat transfer through a fracture with permeable walls in an EGS. AGEC 2011 16-18 November, Melbourne Australia.

Mohais R, Xu C, Dowd PA (2011b) Fluid flow and heat transfer within a single horizontal fracture in an Enhanced Geothermal System. J Heat Transfer (Trans. ASME), 133:112603-1–112603-8.

Murayama Y, Thapa RB (2011) Spatial Analysis and Modeling in Geographical Transformation Process, GIS-based Applications. Springer, pp312.

Nelson RA (2001) Geologic Analysis of Naturally Fractured Reservoirs. Gulf Professional Publishing, pp350.

Oda M, Hatsuyama Y, Ohnishi Y (1987) Numerical experiments on permeability tensor and its application to jointed granite at Stripa mine, Sweden. J Geophys Res 92:8037–8048

Odling NE (1991) A "Conductance" Mesh Approach to the Permeability of Natural and Simulated Fracture Patterns. Water Resources Research, 27(10):2633-2643.

Odling NE (1992) Network Properties of a Two-dimensional Natural Fracture Pattern. Pageoph, 138(1):95-114.

Okabe A, Boots B, Sugihara K, Chiu S.N, (2000) Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. John Wiley & Sons, pp683.

Ozkaya SI, Mattner J (2003) Fracture connectivity from fracture intersections in borehole image logs. J Computer & Geosciences, 29:143-153.

Paluszny A, Zimmerman RW (2011) Numerical simulation of multiple 3D fracture propagation using arbitrary meshes. J Computer Methods in Applied Mechanics and Engineering, 200:953-966.

Papazov C, Burschka D (2010) An efficient RANSAC for 3D object recognition in noisy and occluded scenes. in proceedings of the 10th Asian Conference on Computer Vision (ACCV'10).

Pardo-Igúzquiza E, Dowd PA (2003) CONNEC3D: a computer program for connectivity analysis of 3D random set models. J Computers & Geosciences, 29(6):775-785.

Parisi A, Caldarelli G, Pietronero L (2000) Roughness of fracture surfaces. Europhysics Letters, 52(3):304-310.

Persson PO (1997) Mesh Generation for Implicit Geometries. PhD Thesis, Massachusetts Institute of Technology, pp126.

Preparata FP, Shamos MI (1985) Computational Geometry. Springer, pp000.

Priest SD (1993) Discontinuity Analysis for Rock Engineering. Chapman & Hall, UK, pp473.

Priest SD, Hudson JA (1981) Estimation of discontinuity spacing and trace length using scanline surveys. Intl. J Rock Mechanics and Mining Sciences, 18:183–197.

Rawnsley K, Wei L (2001) Evaluation of a new method to build geological models of fractured reservoirs calibrated to production data. Pet Geosci 7:23–33

Renard P, Allard D (2011) Connectivity metrics for subsurface flow and transport. J Advances in Water Resources, DOI 10.1016/j.advwatres.2011.12.001.

Renshaw CE (1996) Influence of sub-critical fracture growth on connectivity of fractures networks. J Water Resources Research, 27:2633-2643.

Riley MS (2005) Fracture trace length and number distribution from fracture mapping. J Geophysical Research, 110:B08414.

Robert C, Casella G (1999) Monte Carlo Statistical Methods. Springer, pp646.

Robinson PC (1983) Connectivity of fracture systems: a percolation theory approach. J Physics A, 16:605-614.

Ross SM (2007) Introduction to Probability Models. Academic Press-Elsevier, USA, pp801.

Roth G, Levine M.D (1993) Extracting geometric primitives. CVGIP: Image Understanding, 58(1):1–22.

Rouleau A, Gale JE (1985) Statistical characterization of the fracture system in the Stripa granite, Sweden. J Rock Mechanics & Mining Science 22:353-367.

Rouleau A, Gale JE (1987) Stochastic discrete fracture simulation of groundwater flow excavation in granite into an underground. J Rock Mechanics & Mining Science 24(2):99-112.

Sadoyan H, Zakarian A, Avagyan V, Mohanty P (2006) Robust uniform triangulation algorithm for computer aided design. J Computer-Aided Design 38:1134-1144.

Sahimi M (1993) Flow phenomena in rocks: from continuum models to fractals, percolation, cellular automata, and simulated annealing. J Reviews of Modern Physics, 65(4):1393-1534.

Sarkar S, Toksoz MN, Burns DR (2004) Fluid flow modeling in fractures. Massachusetts Institute of Technology, Earth, Atmospheric and Planetary Sciences, Cambridge.

Sausse J, Dezayes C, Dorbath L, Genter A, Place J (2010) 3D model of fracture zones at Soultz-sous-Forets based on geological data, image logs, induced microseismicity and vertical seismic profiles. Comptes Rendus Geosci 342:531–545

Seifollahi S, Dowd PA, Xu C, Fadakar-A Y (2013) A Spatial Clustering Approach for Stochastic Fracture Network Modelling. J Rock Mechanics and Rock Engineering, DOI 10.1007/s00603-013-0456-x.

Serra J (1983) Image Analysis and Mathematical Morphology. Orlando, FL, USA, Academic Press.

Shapiro SA, Royer JJ, Audigane P (1998) Estimating the permeability from fluid-injection induced seismicity emission, Poromechanics, Thimus et al. (eds), Balkema, Rotterdam, 301-305.

Sims DW, Morris AP, Ferrill DA, Sorkhabi R (2005) Extensional fault system evolution and reservoir connectivity. in: Sorkhabi R, Tsuji Y (eds.) Faults, Fluid Flow and Petroleum Traps: AAPG Memoir 85:79-93.

Singhal BB, Gupta RP (2010) Applied Hydrogeology of Fractured Rocks. Springer, pp408.

Snow DT (1965) A Parallel Plate Model of Fractured Permeable Media. PhD Thesis, University of California, Berkeley.

Staub I, Fredriksson A, Outters N (2002) Strategy for a Rock Mechanics Site Descriptive Model. Golder Associates, Sweden, pp219.

Stauffer D, Aharony A (1992) Introduction to Percolation Theory. Taylor and Francis, London, pp181.

Stauffer D, Aharony A (2003) Introduction to Percolation Theory. Taylor & Francis, pp179.

Stone CA, Kuszmaul JS, Boontun A, Young D (1996) Comparison of an analytical and a numerical approach to probabilistic keyblock analysis. in: Aubertin M, Hassani F, Mitri H (eds.) Rock Mechanics: Tools and Techniques, 2:1769-1775.

Stoyan D, Kendall WS, Mecke J (1995) Stochastic Geometry and its Applications. John Wiley, England, pp436.

Szeliski R (2010) Computer Vision: Algorithm and Applications. Springer, 798p.

Tang C, Kaiser PK, 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.

Toth TM, Vass I (2011) Relationship Between the Geometric Parameters of Rock Fractures, the Size of Percolation Clusters and REV, J Mathematical Geosciences, 43:75-97.

Toussaint GT (1983) Computing largest empty circles with location constraints. J Computer & Information Sciences, 12(5):347-352.

Toussaint GT (1985) A simple linear algorithm for intersecting convex polygons. J The Visual Computer, 1:118-123.

Tran NH (2007) Simulated annealing technique in discrete fracture network inversion: optimizing the optimization. Computer & Geosciences, 11:249-260.

Unwin A, Theus M, Hofmann H (2006) Graphics of Large Datasets: Visualizing a Million. Springer, 271p.

Velic M, May D, Moresi L (2009) A Fast Robust Algorithm for Computing Discrete Voronoi Diagrams. J Mathematical Modeling and Algorithms, 8:343–355.

Vince J, 2005, Geometry for Computer Graphics, Springer, London, p359.

Vogel H (2002) Topological Characterization of Porous Media. in: Mecke K.R, Stoyan D (eds.), LNP 600, Springer, p75-92.

Wand MP, Jones MC (1995) Kernel Smoothing. Monographs on Statistics and Applied Probability, Chapman & Hall.

Warburton P (1980) Stereological interpretation of joint trace data: influence of joint shape and implications for geological surveys. Int. J Rock Mechanics & Mining Science, 17:305-316.

Wells NA, (2000) Are there better alternatives to standard rose diagrams? J Sedimentary Research, 70(1):37-46.

Weltner K, Grosjean J, Weber WJ, Schuster P (2009) Mathematics for Physicists and Engineers, Fundamentals and Interactive Study Guide. Springer, pp596.

Wen R, Sinding-Larsen R (1997) Stochastic modeling and simulation of small faults by marked point processes and kriging. in: Baafi, Schofield (eds.) Geostatistics Wollongong 96, Kluwer Academic Publisher, 1:398-413.

Willis-Richards J, Wallroth T (1995) Approaches to the modelling of HDR reservoirs: a review. J Geothermics, 24(3):307-332.

Windsor CR, Thompson AG (1996) Block theory and excavation engineering. in: Aubertin M, Hassani F, Mitri H (eds.) Rock Mechanics: Tools and Techniques, 2:1753-1760.

Wyborn D, de-Graaf L, Hann S (2005) Enhanced geothermal development in the Cooper Basin area, South Australia. Transactions, Geothermal Resources Council, 29:151-156.

Xu C, Dowd PA (2010a) A new computer code for discrete fracture network modelling. J Computer & Geosciences, 36:292-301.

Xu C, Dowd PA, Mardia KV, Fowell RJ (2003a) Spatial Point Density Estimation. Technical Report, Department of Mining and Mineral Engineering, The University of Leeds, pp63.

Xu C, Dowd PA, Mardia KV, Fowell RJ (2003b) Parametric point intensity estimation for stochastic fracture modelling. J Leeds University Mining Association, 16:85-93.

Xu C, Dowd PA, Mardia KV, Fowell RJ (2006) A Connectivity Index for Discrete Fracture Networks. J Mathematical Geology, 38(5):611-634.

Xu C, Dowd PA, Mardia KV, Fowell RJ, Taylor CC (2007) Simulating correlated marked point processes. J Applied Statistics, 34(9):1125-1134.

Xu C, Dowd PA, Mohais R (2013a) Connectivity analysis of the habanero enhanced geothermal system. in proceedings of 37th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, SGP-TR-194.

Xu C, Dowd PA, Wyborn D (2010b) Optimised fracture network model for Habanero reservoir. in: Proceeding of Australian Geothermal Conference AGEC2010, Australia, pp6.

Xu C, Dowd PA, Wyborn D (2011) Optimisation of a stochastic rock fracture model using Markov Chain Monte Carlo simulation. Proceedings of the 35th APCOM Conf, AusIMM, Melbourne, Australia, 635-642.

Xu C, Dowd PA, Wyborn D (2013b) Optimization of a stochastic rock fracture model using Markov chain Monte Carlo simulation, Mining Technology (in press). doi:10.1179/1743286312Y.

Xu C, Jia X, Williams RA (2005) Use of Lattice Boltzmann Method (LBM) to facilitate characterization of pore networks in porous media. in: Proceedings of Particulate Systems Analysis, The Royal Society of Chemistry, Stratford-upon-Avon, UK.

Yang MY, Wolfgang F (2010) Plane Detection in Point Cloud Data. In Proceedings of the 2nd Int Conf on Machine Control Guidance Bonn, 1:95-104.

Yaniv Z (2010) Random Sampling Consensus (RANSAC) algorithm, a generic implementation. The Insight Journal, 2, http://hdl.handle.net/10380/3223.

Zhang D (2002) Stochastic Methods for Flow in Porous Media. Academic Press, San Diego, CA, pp350.

Zhang L, Einstein HH (2000) Estimating the intensity of rock discontinuities. Intl.J Rock Mechanics & Mining Sciences, 37:819-837.

Zimmerman RW, Bodvarsson GS (1996) Hydraulic Conductivity of Rock Fractures, Transport Porous Media 23:30.