University of Alberta

Geostatistics for Naturally Fractured Reservoirs

by

Eric Barritt Niven

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mining Engineering

Civil and Environmental Engineering

Spring 2013
Edmonton, Alberta

To my wife, Kristen.

Abstract

A common problem in naturally fractured reservoirs (NFRs) is a lack of data caused by few wells; or at least, few wells with core or borehole images. Secondary data (such as seismic) can be used to improve predictions of fracture intensity in between the wells. Common geostatistical techniques for incorporating secondary data rely heavily on the correlation coefficient, which is influenced by outliers and whose uncertainty is usually unknown or not assessed in practice. A novel method is developed for calculating a robust correlation coefficient and propagating uncertainty in the correlation through reservoir modelling of fracture intensity. Discrete fracture networks (DFNs) are created to reproduce the models of fracture intensity.

Current DFN modelling techniques incorporate and honour some geological information such as intensity and orientation data. However, most DFN modelling algorithms and software do not account for similarity in the orientation of nearby fractures, fracture network connectivity or fracture spacing in an explicit manner. This thesis shows that some natural fracture networks are not realistically modelled by conventional techniques. A new discrete fracture network simulation algorithm is developed, which works by simulating more fractures than are required and iterating to find a subset that best matches target spatial statistics. It is shown that the proposed simulation algorithm results in fracture networks that are more geologically realistic compared with the traditional methods. The increase in geological realism is expected to lead to better resource predictions and economic decisions for reservoir management.

Acknowledgements

I would like to thank Dr. Clayton Deutsch for his guidance and encouragement throughout the course of my degree. I never once left a meeting with Dr. Deutsch without feeling re-energized about my research.

I would also like to express my gratitude to the member companies of the Centre for Computational Geostatistics for their financial support. I would also like to thank the students of the CCG for their friendship.

Finally, I would like to thank my wife, Kristen, for putting up with me during the final year of my degree. Your support and love has meant the world to me.

Contents

List of Tables

List of Figures

List of Symbols and Nomenclature

κ	A measure of dispersion for the Fisher distribution. Analogous to the standard deviation in the Gaussian distribution
σ	Sigma factor (or shape factor) accounts for transfer between the matrix and fracture network
e	aperture
k_{ij}	permeability tensor
P_{32}	fracture intensity: area of fractures per volume of rock; has units of m^{-1}
CCK	collocated co-kriging
DFN	discrete fracture network
FMF	fracture multiplication factor
FMI	formation micro-images
GRF	Gaussian random function
LOOT	leave-one-out test
LXOT	leave-x-out test
NFR	naturally fractured reservoir
PDF	probability density function
PDF	probability distribution function
PGS	pluri-Gaussian simulation
SGS	sequential Gaussian simulation
SIS	sequential indicator simulation

Chapter 1

Introduction

This chapter is organized as follows. Section 1.1 motivates geological modelling of petroleum reservoirs. Naturally fractured reservoirs are introduced as a special case of petroleum reservoirs and their importance is noted. Section 1.2 provides background on current methods for characterizing naturally fractured reservoirs. Section 1.3 and 1.4 describe the problem statement and summarize the contributions of this thesis. Section 1.5 provides an outline for the rest of the thesis.

1.1 Research Motivation

The proposed research is in the field of geostatistics, which can be thought of as a philosophical approach to spatial data in the presence of natural variability or as a collection of mathematical and statistical techniques that can be used to analyse and model data with spatial locations. Geostatistics was developed in the mining industry and has been used to evaluate natural resources such as mineral or petroleum deposits as well as environmental sites. Geostatistics is often used to make predictions about the value of a particular property, such as porosity, mineral grade, or lithology, at unsampled locations in space. One of the main advantages of geostatistics is that it enables the quantification of uncertainty in our predicted values.

Geostatistical modelling of petroleum reservoirs is now commonplace in a variety of reservoir types, as is evidenced by the wide array of books and articles available on the subject. To name just a few, the reader is referred to Isaaks and Srivastava (1989); Goovaerts (1997); Deutsch and Journel (1998); Deutsch (2002); Wackernagel

(2003); Chiles and Delfiner (2012) and the references contained therein. There are many reasons for the widespread use of geostatistical techniques to build reservoir models. Some important reasons include:

- Geologists and engineers are being faced with an increasing amount of data that occur at a variety of scales (i.e. consider the variation in scale of core plugs, well log data and seismic data). Geostatistical techniques provide a quantitative, unambiguous and repeatable means for generating reservoir models that honour and integrate various sources of data including (but not limited to) core and well data, seismic data, trends and expert geological knowledge (Deutsch, 2002).
- In addition to providing estimates or simulated values of reservoir properties, geostatistical models provide a means for assessing uncertainty in the modelled resource. Kriging provides the variance of local estimates while stochastic simulation enables assessment of global uncertainty in reservoir properties. For example, the geomodeller can calculate maps or volumes of the probability of shale at each grid cells in a model. Such a map would be useful for locating new wells.
- It is also important to provide a reliable means for calculation of resource estimates. These volumetric estimates can be used to project the economic value of reservoirs and can be used as a basis for comparing alternative resource projects.

For those interested readers, Deutsch (2002) provides additional discussion motivating geostatistical reservoir modelling.

This thesis is mainly concerned with geostatistical modelling of naturally fractured reservoirs (NFRs). A **NFR** is defined as “a reservoir in which naturally occurring fractures either have, or are predicted to have, a significant effect on reservoir fluid flow, either in the form of increased reservoir permeability and/or reserves or increased permeability anisotropy” (Nelson, 2001). It’s important to note that fractures may not only provide a positive effect on the flow performance of a reservoir. In the case where fractures are in-filled with gouge, mineralization or another low permeability substance, they may act as barriers to flow. A **reservoir fracture** is a “naturally occurring

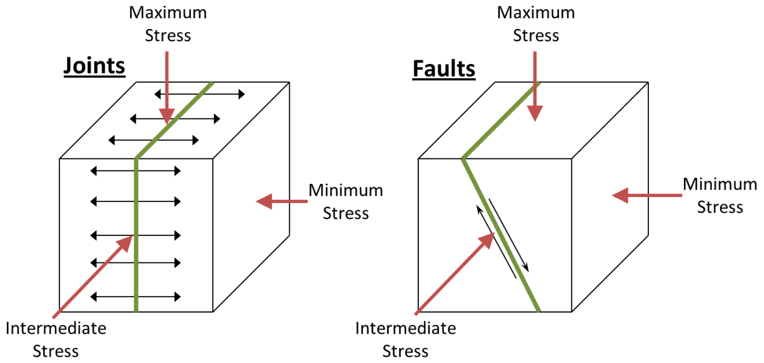


Figure 1.1: Two common fracture styles; joints and faults. The relative orientation of the principal stresses with respect to the fracture plane is shown along with the direction of displacement (Modified from Narr et al. (2006)). Not to scale.

macroscopic planar discontinuity in rock due to deformation or physical diagenesis" (Nelson, 2001). There are two types of fractures that fit this definition: faults and joints (U.S. Department of the Interior Bureau of Reclamation, 1998; Park, 2004). Figure 1.1 shows the formation of joints and faults relative to their stress state.

Faults are created oblique to the direction of the principal stresses at the time of faulting and show significant shear displacement. Joints, on the other hand, are fractures with no shear displacement and small displacement normal to their surfaces. Joints develop parallel to the maximum horizontal stress at the time of formation (Narr et al., 2006) (See Figure 1.1).

Rock fractures are prevalent in the Earth's upper crust. They are seen in nearly all rock masses by observing fractures in core and most rock outcrops. Petroleum reservoir rock is no different. It is likely that most petroleum reservoirs contain fractures (Narr et al., 2006). In some reservoirs, the fractures do not play a significant factor in the development and production of hydrocarbons. In others, fractures can play a key role in production either by enhancing fluid flow (in some cases making production from an otherwise low-quality reservoir feasible) or by acting as a barrier.

Reliable estimates on how much of the world's petroleum reserves are known to be in NFRs are hard to come by, but some estimates in the literature are as high as 60% (Beydoun, 1998; Roxar, 2009; Waldren and Corrigan, 1985).

In order to optimize management and production of NFRs, detailed information on the attributes, properties and behaviour of the fracture network must be known along with those of the rock matrix. There are several aspects unique to NFRs that make them more difficult to characterize, model and produce compared to other reservoirs whose fractures have little or no effect on production.

- In NFRs there is a need to characterize the fractures in addition to the rock matrix. Characterization of the fractures involves gathering data to define fracture attributes such as size, orientation, spacing, porosity, aperture, intensity and connectivity. Some of these attributes, such as aperture and size are difficult to characterize (see Section 1.2).

- One of the difficulties in characterizing the reservoir fractures is simply in gathering enough data. Lack of sampling is a common problem in NFRs where there may be few vertical wells with available core or image log data from which to draw fracture statistics. Further complicating matters is the fact that vertical wells have a relatively low probability of intersecting sub-vertical fractures (Baker and Kuppe, 2000; Narr et al., 2006; Makel, 2007). Cored horizontal wells provide a valuable source of additional information on vertical fractures, but are not normally available.

- There is also a need to characterize the matrix-fracture interactions. The interaction between the high pore volume but low permeability rock matrix and the low pore volume but high permeability fractures is a function of matrix architecture, fracture geometry, stress state and the mechanisms and physical processes that control the transfer of hydrocarbons from the matrix to the fractures (Makel, 2007).

- Directly measured fracture data only exist at well locations. Unfortunately interwell predictions of fracture properties is difficult due to their complexity and high variability. Fracture porosity and fracture permeability may show shorter correlation ranges than their matrix property counterparts (Makel, 2007).

Although developing permeability models to represent NFRs is a challenging task due to the aforementioned reasons, the uncertainty in predictions of reservoir performance can be reduced by integrating as much available data as possible (Reza, 2003). As such, there is a need to integrate all available information such as: formation micro-images (FMI), well-test data, production data, seismic surveys, well log data and data from outcrops. This thesis proposes novel approaches to the integration of multiple data types into geomodels of NFRs.

1.2 Characterization and Modelling of NFRs

Characterizing and modelling NFRs is challenging; however, the task can be distilled down into a few main goals: 1) observing the fractures in the wells and determining which (if any) are important; 2) determining reasonable fracture attributes; 3) determining how best to model the inter-well locations; and 4) translating and simplifying fracture information for a flow simulator.

1.2.1 Data Sources for Characterization of NFRs

There are several potential sources for data that may help in the characterization of natural fracture networks such as: core data, image logs, borehole logs, seismic geophysics, outcrop analogues, production tests and production data. Core data is the most useful data source, especially when oriented, and is the only way direct way to measure fractures in a reservoir. Other sources of data have varying degrees of usefulness. For example, seismic data may be very useful in some reservoirs and not others. Production testing could be useful to determine the permeability of a fault, which cannot be known from core data.

Sampling bias due to the orientation of the borehole in relation to the fracture orientation is common, but can be corrected for using the Terzaghi correction (Terzaghi, 1965), which is discussed in more detail in Chapter 4. Estimates of fracture apertures from core are highly uncertain due to removing the core from its in-situ stress state and possible core damage due to drilling and handling. However, it may be possible to estimate apertures by measuring the width of joint fill in the case of cemented or partially cemented joints (Makel, 2007). Making the distinction between shear and tensile fractures (joints) is only possible by examining core and can be important if the two different types of fractures have different dimension characteristics and spatial distributions (Cacas et al., 2001; Odling et al., 1999). Fracture type can also be used to help determine relative age of fractures and split them into sets.

Borehole image logs yield similar information compared to core data and is also very useful. One advantage of image logs is that they provide a more complete record compared to core since they are often taken over greater depth intervals. However, resolution limitations may mean that some fractures with small apertures are not detected. It is possible, in theory, to determine apertures from image logs; however, accuracy is limited by borehole image resolution (Luthi and Souhaite, 1990). Even though core and image logs provide similar information, they can be used together to reduce uncertainty in the measured fracture attributes.

If suitable outcrops are available, they may provide both qualitative and quantitative information. Outcrops can help define the fracturing style (dispersed systematic joint sets vs. localized swarms of fractures) as well as provide information on the orientation, length and spacing of fractures (Cacas et al., 2001). Outcrop data should be used with extreme caution since they are located some distance away from the actual reservoir and are subject to stress unloading and weathering.

Seismic geophysical surveys can be used to identify the location of large-scale faults and fractures. Seismic amplitude attributes, such as coherency, can be analysed and used to create fault probability maps (Bourbiaux et al., 2002). Seismic anisotropy, or azimuthal, analysis is a relatively new development that shows promise for mapping fracture orientation and intensity between wells (Bourbiaux et al., 2005; Angerer et al., 2004). It works by decomposing the seismic attributes into a common geological part, an azimuth-dependent part (which supplies the fracture information) and random noise. Sometimes seismic AVO or shear wave attribute maps can be generated to infer small-scale fracture orientation and density changes (Eikmans et al., 1999). Seismic data must be corroborated with well bore information to assess the validity of using it to predict at inter-well locations.

Drawdown or buildup tests can be conducted. In a draw-down test, the well is pumped for some time and then shut-in. The pressure response and fluid level in the well is monitored over time. A plot of log-time versus log-pressure can provide an indication of the effective fractures and fracture connectivity (Narr et al., 2006). Well tests can also provide an indication of both a fracture dominated response and a matrix dominated response, which is called a dual porosity system; however, many NFRs do not exhibit the characteristic dual porosity pressure response.

Interference testing between multiple wells can provide an assessment of reservoir flow over larger areas (Narr et al., 2006). This is done by creating a pressure pulse in one well and measuring the response in nearby wells and can provide valuable information about large scale flow anisotropies. Tracer tests can also be used whereby a unique tracer is injected in one well and its presence is monitored for in nearby wells.

Historical production data can be integrated into geomodels of petroleum reservoirs (whether or not they are considered NFRs). Numerical history matching of production data is an inverse problem where the geological model is modified in order to minimize the mismatch between the simulated and historical production profiles. The model parameters that could be modified for history matching are mainly porosity and permeability, but could also include hydrocarbon volume, compressibility, vertical to horizontal permeability ratio and reservoir geometry. Unfortunately, numerical history matching is a difficult and computationally expensive task. Although automatic history matching algorithms exist, much of the history matching occurring in practice is due to manual iteration of geological models (Deutsch, 2002).

1.2.2 Fracture Data Analysis and Statistics

An important task in any reservoir analysis is to gather and analyse all available data. For NFRs, there is the additional task of compiling information on the fracture network. It is important to define distributions of fracture attributes such as orientation, spacing, length, porosity, aperture and connectivity, which are a function of the reservoir rock porosity, lithology, structural position and brittleness (Baker and Kuppe, 2000).

Orientation

Stereonets and rose diagrams are used for visualizing and analyzing fracture orientation data. Stereonets make it easy to group fracture data into sets based on similar orientations and/or fracture type. Splitting the data into sets may be harder in some fields than in others. If the average fracture orientation varies with location, it may be hard to separate the data into sets. It may be helpful to split the reservoir into separate regions before plotting separate stereonets for each region. The joint orientations on a stereonet can be contoured and an average orientation could be read directly from the plot (visually) if only an approximate value is required. For additional accuracy, the Fisher distribution is often used to characterize fracture data sets. The Fisher distribution is analogous to the normal distribution on a sphere (Davis, 2002). The Fisher distribution is parameterized by angles ϕ and θ, which are measured from the z and x axes, respectively, and a concentration parameter, κ (See Equation 1.1).

$$f(\phi', \theta') = \frac{\kappa \sin\theta' e^{\kappa cos\phi'}}{2\pi(e^{\kappa} - 1)}; 0 \le \phi' \le 2\pi \tag{1.1}$$

The Fisher distribution is usually valid if $\kappa > 5$. When the number of fracture poles is greater than 30, κ can be estimated as follows:

$$\kappa = \frac{N_f}{N_f - |R|} \tag{1.2}$$

where $|R|$ is the magnitude of the vector sum of the unit vectors for orientation (Golder Associates Ltd., 2010).

Figure 1.2 shows an two examples of fracture poles generated using a Fisher distribution. The figure shows that as κ increases, the spread of the distribution decreases. Uniform dispersion across the sphere can be specified by setting $\kappa = 0$.

Length and Shape

Fracture lengths in a single set may vary over several orders of magnitude. Micro fractures down to the millimetre scale may exist along with very large fractures or faults that are kilometres in length (Twiss and Moores, 1992).

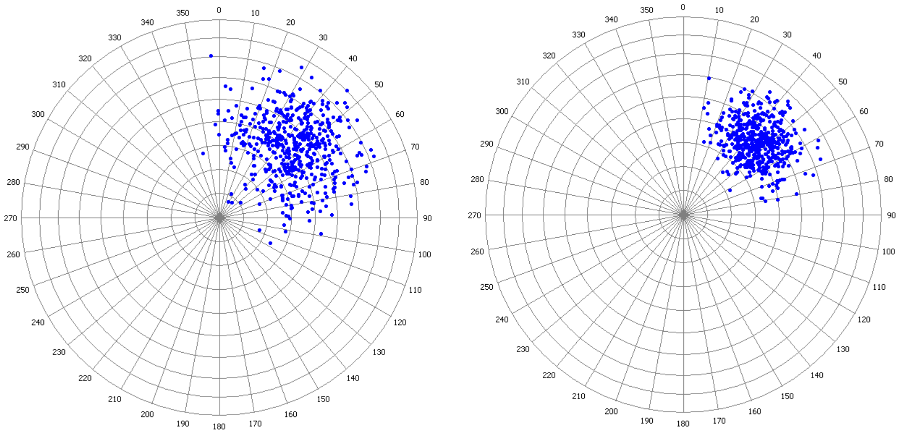


Figure 1.2: Fracture poles simulated using the Fisher distribution for two values of κ. On left: $\kappa = 20$. On right: $\kappa = 50$.

Fracture shape depends on lithology and geological structure (i.e. bed thickness). While fractures in granite may be nearly elliptical in shape, they are often more rectangular in bedded sandstones or shales. The reason is generally due to fracture growth being constrained by upper and lower bedding contacts. In these cases fractures may tend to be longer parallel to bedding than across it as their growth often terminates at the bed boundaries (Twiss and Moores, 1992).

Aperture

Evaluation of fracture aperture data is extremely important. Parallel plate theory indicates that flow through a fracture is related to the cube of aperture (Witherspoon et al., 1980; Koudina et al., 1998; Hosseinian et al., 2010).

$$Q = \frac{e^3 \Delta p}{12 \mu L} \tag{1.3}$$

where Q is the volumetric flow rate, e is the aperture, Δp is the pressure drop, L is the distance between the inlet and outlet boundaries, and μ is the fluid viscosity. Thus, some indication of aperture is extremely important for predicting fracture permeability. The distribution of fracture apertures is generally thought to be log-normal (Hakami and Barton, 1990; Iwano and Einstein, 1993; Johns et al., 1993; Pyrak-Nolte et al., 1997).

As noted earlier, fracture aperture can be measured directly from core. Although these measurements are subject to high uncertainty, it is preferred to no measurements (Narr et al., 2006). Where resistivity image logs are available, apertures should be computed and compared to those from core. The aperture data from resistivity logs is uncertain, at least in terms of absolute aperture values. However, the relative size of the aperture sizes are often consistent with those from core and therefore can be used as additional information to help validate the core measurements (Narr et al., 2006; Luthi and Souhaite, 1990). If a correlation exists between apertures from core and those from FMI, this correlation can help assess apertures in wells that only have FMI available.

Mud loss data can be useful for evaluating fracture apertures. Mud losses that build gradually are thought to occur as mud is lost to the permeable matrix of the borehole wall. Mud losses to open fractures may occur abruptly. It is, therefore, possible to estimate fracture aperture and size through highly-accurate monitoring of mud losses (Verga et al., 2000).

Spacing, Density and Intensity

Spacing, density and intensity are interlinked fracture network attributes. Fracture spacing is the orthogonal distance between fracture planes and can be measured as the distance between fractures along a scan line, or down a borehole (Makel, 2007; Twiss and Moores, 1992). Measured spacings may need to be corrected since they are dependent upon the relative orientation between the sampling line and the fracture orientation. The Terzaghi correction (Terzaghi, 1965) is the most common way to correct spacing data (See Chapter 4 for more details).

Spacing

Fracture spacing data has been presented many times in the literature. It is widely observed that joint spacing is proportional to bed thickness (Narr and Suppe, 1991; Ji and Saruwatari, 1998; Wu and Pollard, 1995; Price, 1966), although very thick beds may depart from this observation (Ladeira and Price, 1981; Narr and Suppe, 1991).

Priest and Hudson (1976) and Wines and Lilly (2002) both presented data showing that fracture spacings followed the negative exponential distribution. However, Priest and Hudson (1976) note that the negative exponential distribution does not apply if there is a predominance of evenly spaced fractures. Other researchers have found that the fracture spacing distribution follows a log-normal distribution (Narr and Suppe, 1991; Bridges, 1975; Becker and Gross, 1996). Huang and Angelier (1989) present data from the Gulf of Suez and south-eastern France where the joints followed a Gamma distribution, which they note is similar to a log-normal distribution. Ji and Saruwatari (1998) presented joint spacing data from three beds that was normally distributed in two beds, indicating roughly regular spacing, and positively skewed in the third bed, which could be described by either the log-normal or gamma distribution.

It seems that spacing distributions described by log-normal or gamma distributions are a consequence of the stress relaxation around a joint, which prevents formation of new joints within some distance from existing joints. The stress relaxation shadow is proportional to the joint height and explains why there is a correlation between bed thickness and joint spacing (Becker and Gross, 1996).

The terms fracture density, intensity and porosity are similar to each other, but are not equivalent. The difference between the three measures is related to the dimension of the samples and the dimension of the measurement region (see Table 1.1). The format for describing fracture intensity (and density and porosity) are based upon the designation P (for persistence), followed by subscripts for the dimensions of the measurement region and the fracture sample, respectively. Thus, a P_{32} fracture intensity refers to the total area of fractures per volume of rock and is the most commonly used measure when modelling fractures in three dimensions (Schlumberger, 2007). It is usually recommended to calculate a fracture intensity measure that is related to the dimension of the modelling area (Dershowitz and Herda, 1992).

1.2.3 Geostatistical Reservoir Modelling

Geostatistics was introduced briefly in Section 1.1 but further discussion is warranted since geostatistical modelling of NFRs presents some unique challenges. This subsection

Table 1.1: Measures of fracture density, intensity and porosity (modified from Dershowitz and Herda (1992)).

		Dimension of Measurement				
		0	**1**	**2**	**3**	
Dimension of Sample	**1**	P_{10} No of fractures per unit length of borehole	P_{11} Length of fractures per unit length			**Linear Measures**
	2	P_{20} No of fractures per unit area	P_{21} Length of fractures per unit area	P_{22} Area of fractures per unit area		**Areal Measures**
	3	P_{30} No of fractures per unit volume		P_{32} Area of fractures per unit volume	P_{33} Volume of fractures per unit volume	**Volumetric Measures**
		Density		**Intensity**	**Porosity**	

discusses one of the more common reservoir modelling work flows using some of the most popular and well-established geostatistical techniques.

Geostatistical Modelling of Rock Matrix Properties

The following suggested reservoir modelling work flow follows that of Deutsch (2002) but incorporates additional details and the author's own personal experience.

The first step in any reservoir modelling work-flow is to collect the relevant data for preliminary data analysis and checking. Types of data that may be used in the model are largely dependent on what is available. Generally the geomodeller should wish to incorporate as much information as possible into the model. Typical data types incorporated in a petroleum reservoir geomodel include: well logs, core analysis, core photos, image logs, horizons identified from seismic surveys, well tops or picks made by geologists, seismic attributes, well-test data and production data.

A visual inspection of the data is helpful. Viewing the well log data is often necessary to identify obvious problems. Histograms and scatter plots of the variables of interest are useful for identifying outliers. Determining statistically representative facies proportions and rock properties is critical. Declustering should be considered to arrive at representative facies proportions and rock properties (Deutsch, 2002).

Establishing the stratigraphic layering involves developing a conceptual model and stratigraphic framework that define the geometry and stratigraphy of the reservoir. Practically speaking, this involves zoning the reservoir into different stratigraphic or geologic units or layers, which form the large-scale reservoir architecture. Each layer corresponds to a particular time period and the surfaces (or horizons) that separate these layers correspond to a significant geologic change (Deutsch, 2002). The idea is to sub-divide the reservoir into large-scale geologically similar units. Most often, the boundaries or horizons between layers are “picked” from well logs and/or by looking directly at the core. Horizons are usually estimated at inter-well locations using one of any number of mapping techniques such as: kriging, inverse distance, spline methods, etc. Some horizons can be mapped from two or three-dimensional seismic surveys.

Modern day commercial reservoir geomodelling software is generally capable of defining complex corner-point grids, called stratigraphic grids, that conform to faults and undulating horizons. Grids could be constructed to conform to a maximum flooding surface. The grid for each geologic layer could conform to separate horizons. It may be desirable to construct a grid parallel to some surface that has since been eroded, rather than present bounding surfaces. Stratigraphic grids are usually transformed to a Cartesian grid (where cell has an i,j,k index) for easier geostatistical calculations.

Facies rock types are modelled before rock matrix properties such as porosity since petrophysical properties are often highly correlated with geological facies. There are any number of ways of defining facies and that discussion is outside the scope of this thesis. In siliclastic petroleum reservoirs it is common to distinguish between sandstone and shale while in carbonate reservoirs it is common to distinguish between limestone and dolomite (Deutsch, 2002).

There are generally two main different approaches to rock type modelling: 1) cell-based approaches and 2) object-based approaches.

Sequential indicator simulation (SIS) is an example of a cell-based approach and is common (Deutsch, 2002) in petroleum reservoir geomodelling. SIS involves:

1. considering K mutually exclusive facies categories and K indicator transforms at

each data location;

2. visiting a grid node location;
3. finding nearby data (including previously simulated grid nodes);
4. constructing a conditional distribution by indicator kriging (which amounts to calculating the probability of each facies being present at that location); and
5. randomly drawing a simulated facies from the set of probabilities.

There are significant advantages to SIS. The facies proportions are honoured and the sample data are reproduced. The main drawback to SIS is that the results do not appear geologically realistic. While the facies proportions and the variogram are honoured in SIS, higher order statistics are not. As a result, there is no way to define or calculate realistic geologic shapes that are seen in petroleum reservoirs.

Truncated Gaussian and pluri-Gaussian simulation are alternatives to SIS. These methods are not seen as often in practice since they require a clear ordering of the facies, which may be difficult to define.

In truncated Gaussian simulation, realizations of a Gaussian random field (GRF) are generated. Cutoffs are applied to the realizations to create categorical facies realizations. For example, in model with three rock types, Gaussian values below -1 might be assigned to facies code 1, values between -1 and 1 might be assigned facies code 2, and values greater than 1 might be assigned facies code 3. Since the Gaussian random field has some spatial continuity, it would be rare to have facies 1 directly next to facies 3. Usually there would be a transition from facies 1 to 2 to 3 and vice versa.

Truncated pluri-Gaussian (PGS) simulation is a generalization of truncated Gaussian simulation that uses two GRFs (Y_1, Y_2) instead of one. PGS requires the definition of a facies substitution diagram. The horizontal axis represents the transform of one GRF, while the vertical axis represents the transform of the other GRF. The diagram shows which facies will prevail at a location given the values from the two GRFs (y_1, y_2). The areas associated with the facies correspond to their proportions if Y_1 and Y_2 are independent (Chiles and Delfiner, 2012). PGS may be difficult to use, in practice, due to the difficulty of defining an appropriate substitution diagram.

The multiple point statistics (MPS) approach was introduced as an alternative to the classical categorical methods such as SIS or PGS. Multiple point statistics include probabilities involving more than two locations. MPS methods use a multiple point histogram and a training image instead of a variogram to estimate conditional probabilities at simulation locations given the observed and previously simulated data. The multiple point histogram is the probability of all combinations of a set of indicators at many locations simultaneously (Lyster, 2009). The training image is a conceptual model of the random process that is representative of the geology of the area of interest (Journel and Zhang, 2006). MPS methods have the advantage of generating more geologically realistic results. However, there are challenges in honouring dense well data. There is also the issue of how to generate or select a training image (Lyster, 2009).

Object-based or process-mimicking techniques attempt to mimic idealized facies body geometries that are interpreted from outcrops and modern analogues (Deutsch, 2002). Object-based models may look more geologically realistic in the sense that they are visually attractive and show non-linear continuity, which cannot be modelled by other cell-based approaches such as SIS. These approaches place geometric objects at locations in space until well data and target facies proportions are honoured. Process-mimicking methods attempt to model the true geologic progression by populating older objects first followed by erosion and deposition of new objects until the model is filled. Object-based and process-mimicking techniques require definition of many parameters and have difficulty honouring dense well data (Lyster, 2009).

Regardless of the method chosen for rock type modelling, all available information should be incorporated into the model. If available, seismic data can be incorporated (Deutsch, 2002) and can add significant information since it is areally extensive over the reservoir. It is also common to condition the models to vertical proportion curves, which show how facies proportions vary with depth in the reservoir (Deutsch, 2002).

Once the rock type model has been simulated, petrophysical properties are simulated within each rock type. Rock matrix properties like porosity, water saturation, oil saturation, the volume fraction of shale and permeability are often modelled. Porosity is measured via well logs, while permeability is measured in the lab on core samples. Thus, there is usually far more porosity data than permeability data. For this reason, permeability is usually modelled after porosity using the measured relationship between porosity and permeability.

There are a number of methods that could be used to create realizations of petrophysical properties. Sequential Gaussian simulation (SGS) (Deutsch, 1992), spectral simulation (Chiles and Delfiner, 2012) and turning bands (Chiles and Delfiner, 2012) are three of the most common. All three methods follow a Gaussian formalism and can be made to honour local data, a histogram and a variogram. The modelling strategy uses the previously defined layering and realizations of rock type to capture abrupt discontinuities. All three Gaussian methods for property simulation require transformation of the original data to the normal distribution. A variogram must be calculated on the normal scored data. SGS proceeds (similarly to SIS) as follows:

1. first, the nearby data are found (including previously simulated nodes);
2. the conditional distribution at that location is calculated by kriging;
3. a simulated value is drawn randomly from that conditional distribution.

The process then repeats, proceeding through a random path of all locations in the model. Multiple realizations are created by changing the random number seed.

Seismic data, such as acoustic impedance, is sensitive to changes in matrix porosity and can be used as an aid in petrophysical property modelling. In order to use the seismic data for this purpose, it must first be calibrated to the well data. If it can be assumed that the porosity and the seismic data follow a bivariate Gaussian distribution, then the correlation coefficient is all that is required to use the seismic data as a predictor for porosity (Deutsch, 2002).

Collocated cokriging (CCK) (Doyen, 1988) can be used to calculate property estimates that consider both well and seismic data. The CCK technique works by kriging with the primary data (i.e. porosity well data) and a single collocated seismic data. The seismic data is weighted via the correlation coefficient (Deutsch, 2002).

Assignment of rock matrix permeability can be made via SGS. An alternative is to transform porosity (which is easier to model given the greater abundance of data compared with permeability) with a calibrated porosity-permeability relationship.

Geostatistical Modelling of Fracture Properties

Analysis of core or borehole image data gives a reasonable idea as to the fracture attributes and flow properties at the well locations. However, some idea of the fracture network characteristics in the inter-well locations is also required. In particular, fracture intensity is required at inter-well locations in order to constrain the discrete fracture network model. Fracture orientation can also be used to further constrain the simulation the of a fracture network. Modelling of attributes such as fracture intensity and orientation typically use the same geo-cellular grid that is used in rock matrix modelling.

Geostatistical methods such as kriging and SGS can be used to populate gridded models of fracture properties (Olarewaju et al., 1997). Some practitioners have used correlations with reservoir characteristics to improve field-wide modelling of fracture intensity (Gauthier et al., 2002; Bourbiaux et al., 2002). For example, if rock matrix porosity is correlated to fracture intensity, the previously generated model of rock matrix porosity can be used to constrain the simulation of fracture intensity through the measured correlation coefficient. Makel (2007) discusses the possibility of correlating horizon curvature and seismic anisotropy with fracture network attributes such as intensity and orientation. Collocated co-kriging or co-simulation is ideal in such a situation.

Coring wells is expensive. In some NFRs, there may not be many wells with direct measurements of fractures from core. Obtaining borehole images is less expensive, but is still often only available in a small subset of wells. In these cases, there may not be very many data with which to calculate a correlation between fracture intensity and some other property or attribute. The correlation coefficient is highly sensitive to outliers (Abdullah, 1990; Isaaks and Srivastava, 1989; Kim and Fessler, 2004; Shevlyakov, 1997) and the uncertainty can be large when the number of data is small (Kalkomey, 1997).

1.2.4 Discrete Fracture Network Modelling

A discrete fracture network (DFN), is a discrete object model of the fractures in a rock mass. The main purpose of fracture network modelling is to study the interconnectedness and hydraulic potential of the fracture network and propagate those properties through flow simulation to examine the effect of the fracture network on the reservoir production (Makel, 2007; Bourbiaux et al., 2005). The measured fracture data used to constrain the DFN are usually split into two categories: 1) large-scale (i.e. reservoir scale) faults and fractures that show up on seismic surveys and 2) small-scale fractures and faults that do not show up on seismic surveys. The large-scale faults and fractures can be modelled as deterministic objects. Small-scale fractures are modelled probabilistically from distributions of fracture attributes defined by the geological data analysis of core, image logs and other sources. A geocellular model of fracture intensity constrains the DFN generation for each fracture set, possibly using the rock matrix geomodel as underlying information. For example, the fracture density of a particular set may be controlled by geomodels of facies and reservoir curvature, while a second set may be controlled by a different facies model and proximity to faults. Since lithology usually plays a significant role in the development of natural fracture networks, the rock type model developed earlier is used to further constrain the DFN modelling (Cacas et al., 2001).

The Baecher model (Baecher et al., 1977; Baecher, 1983), was one of the first models characterizing fracture network generation. Under the Baecher model, fractures centroids are randomly located in space using a Poisson process and the fractures are generated as disks with an orientation and radius. The enhanced Baecher model (Dershowitz, 1988) extends the original Baecher model by providing a provision for fracture terminations and more general fracture shapes. Terminations are specified as a probability that a fracture will terminate at a pre-existing fracture, given that an intersection is detected.

The nearest neighbour model is a non-stationary model in which the fracture intensity (P_{32}) decreases exponentially with distance from major features (faults) that are specified. The P_{32} intensity at any point in space is defined by:

$$P_{32}(x) = Ce^{-bd_{x,f}} \tag{1.4}$$

where C and b are constants and $d_{x,f}$ is the distance between location x and the nearest major feature.

Some researchers have presented data from natural fracture networks that follow fractal patterns. Thus, models to generate fractal fracture network models have been developed. Fractal networks can be simulated by:

1. Producing fracture patterns at one scale. Then superimposing them at different scales to produce a fractal pattern;
2. Generating fractures using non-fractal processes, then test the resulting patterns to determine whether or not they are fractal in nature;
3. Using a process such as a Levy flight, which has been shown mathematically to produce fractal patterns.

The Levy flight process (Mandelbrot, 1982) is a sort of random walk where the length, L, of each step is given by the probability function:

$$P(L > L_s) = L_s^{-D} \tag{1.5}$$

where D is the fractal mass dimension of the point field of fracture centres and L_s is the distance from one fracture to the next for the previous step in the generation sequence. Although the Levy flight is a one dimensional process, it has been extended to two and three dimensions for the generation of fractures in the Levy-Lee fractal model (Golder Associates Ltd., 2010; Clemo, 1994). If $D = 0$, the distribution of step length is uniform (there is no clustering). For larger D values there is low probability of large steps and fractures are found in concentrated clusters. Various researchers have reported fractal dimensions ranging from 0.8 to 2.4 for trace survey maps (Stach et al., 2001; La Pointe, 1988; Ghosh and Daemen, 1993).

Some research has focused on generating more geologically realistic DFNs by growing fractures from an initial seed location. Renshaw and Pollard (1994) simulated fracture networks using geomechanical principles by propagating fracture tips when the stress exceeds a critical threshold. Their approach was successful in yielding realistic looking images of fractures but was computationally prohibitive and limited to two dimensions. Although their work was conducted almost 20 years ago, this type of geomechanical fracture modelling is still too computationally expensive for reservoir-scale three-dimensional modelling. Work by Srivastava (2006) mimics that of Renshaw and Pollard in that fractures are propagated at their tips, but instead of being governed by geomechanical principles, his work is governed by statistical rules. Initial fracture locations are seeded and fracture traces are propagated in two dimensions at the surface by using sequential Gaussian simulation (SGS), which incorporates nearby data into a local distribution of possible azimuths for the next segment. Once the surface traces are simulated, they are propagated to depth, again using SGS, by simulating a dip angle from nearby data. Srivastava's approach was successful in yielding realistic three dimensional fracture networks; however, his work requires detailed joint mappings from outcrops, which are not available in most petroleum reservoirs and are often affected by weathering and stress unloading anyway.

Fracture apertures can be assigned to fractures in a number of different ways. Perhaps the simplest method would be to assign the same aperture value to each of the modelled fractures. A single fixed value could be assigned to all fractures in simple models or apertures could be sampled from different probability density functions (PDFs) for different fracture sets. It is also possible to interpolate gridded fracture porosity values over the geo-cellular grid and assign apertures proportionally to each fracture such that the fracture porosity for the grid cell is honoured (Makel, 2007).

1.2.5 Fracture Network Upscaling

Most current approaches for flow simulation of NFRs rely on either the continuum or the DFN approach. Under the DFN approach, flow is simulated directly on the DFN fractures using an unstructured grid, allowing incorporation of many of the characteristics of real fracture systems. Although the DFN approach can handle complex fracture geometry, its use has typically been limited to basic flow calculations and assumes zero permeability matrix rock. The continuum approach has seen far more use in actual practice (Makel, 2007). Under the continuum approach, the flow on a fracture is not directly simulated. Instead, the DFN is represented by a geocellular grid and an equivalent permeability is calculated for each grid element based on the fractures occurring in that grid cell. The continuum approach has the advantage that it can simulate complex recovery mechanisms such as capillary pressure and matrix-fracture interactions (Rodriguez et al., 2006; Makel, 2007).

One of the most popular methods for calculating an equivalent permeability tensor was developed by Oda (1985). Oda's approach is fast because it does not require flow simulation and it can obtain effective properties directly from the geometry and properties of the fractures within the grid cells.

Oda's approach lays a specified grid (the geocellular grid) on top of a DFN and derives effective properties based on the fractures contained in each cell. The effective permeability for the grid cell is calculated by decomposing the permeability vector into the x, y and z-directions. The permeability tensor is:

$$k_{ij} = \frac{1}{12}(P_{kk}\delta_{ij} - P_{ij}) \tag{1.6}$$

where:

- δ_{ij} is the Kroenecker delta;
- $P_{kk} = P_{11} + P_{22} + P_{33}$;
- $P_{ij} = \frac{1}{V}\sum\limits_{k=1}^{N} l^2 e^3 n_{ik} n_{jk}$;
- V is the grid cell volume;
- l is the length of the fracture;
- e is the fracture aperture; and
- n_{ik} and n_{jk} are the components of a unit normal to the fracture k.

Oda's method is simple and quick to calculate but does not consider the interconnectedness of the fracture network. This assumption is considered acceptable for well connected fracture networks (Dershowitz et al., 2004).

An alternative to Oda's method is flow based upscaling. This method amounts to applying a pressure gradient in one direction (the i, j or k direction) and back calculating permeability. The process is repeated for each direction and for each grid cell. The simulator creates a finite element, unstructured grid along fracture surfaces (Schlumberger, 2007). In this way, flow along the actual fractures is simulated. The advantage of this method is that it directly accounts for the fracture geometry. The disadvantage is that the flow calculations are far slower than Oda's vector addition operations. The flow based method would be impractical for upscaling large reservoirs with many fractures. However, it could be used at the well bore scale, to predict local permeability. It could also be used to validate the results of Oda's method (Schlumberger, 2007).

The important flow properties that must be calculated from the fracture network are the fracture porosity, the directional fracture permeabilities and the sigma factor.

The sigma factor (sometimes called the shape factor), σ, is related to the connectivity between the rock matrix and the fractures and is a function of fracture spacing (Gilman, 2003). The sigma factor is needed by flow simulators to calculate the fluid transfer between the fracture network and the matrix. The pseudo-steady state, analytically derived expression for the shape factor in terms of the mean fracture spacing in the grid cell, L, in the x, y and z directions is (Kazemi and Gilman, 1993):

$$\sigma = \pi^2 \left(\frac{1}{L_x^2} + \frac{1}{L_y^2} + \frac{1}{L_z^2} \right) \tag{1.7}$$

The numerically derived equation for the shape factor is the same except that the π^2 term is replaced with a 4 (Kazemi et al., 1976).

1.3 Problem Statement

As is noted earlier, geostatistical modelling of petroleum reservoirs, in general, and NFRs, in particular, often relies on the correlation coefficient between a primary and a secondary variable. However, the correlation coefficient is known to be highly sensitive to outliers and is uncertain when the number of data is small. A robust measure of correlation is needed along with a methodology for quantifying its uncertainty and propagating that uncertainty through geostatistical calculations.

Current DFN modelling techniques incorporate and honour some geological information about the fractures. Where data is available, fracture lengths and apertures can be honoured. DFNs can also honour fracture intensity and orientation data. Although uncommon, it is possible to impose some control on fracture clustering (not the same as fracture spacing) in the DFN by using a Levy-Lee model. Calibration or trial and error would be required to honour a desired fracture spacing distribution, which would need to be fractal in nature. In many rock masses fractures tend to be oriented similarly to their nearest neighbours. The author is not aware of any fracture simulation algorithms or models that account for similarity in the orientation of nearby fractures, the number of intersections in the fracture network or fracture spacing (in an explicit manner).

Thesis Statement

Geostatistical modelling of naturally fractured reservoirs will be significantly improved by: 1) development of a robust correlation coefficient; 2) development of a methodology for considering the uncertainty in the correlation and propagating its uncertainty through geostatistical modelling; 3) development of a new DFN simulation algorithm that honours geologically realistic spatial statistics.

1.4 Thesis Contributions

This thesis makes several important contributions. Geomodellers commonly study multivariate problems where incorporation of different variables in the geological modelling can improve predictions by considering the correlation between those variables. Traditional measures of correlation are highly sensitive to outlier data. A new robust correlation coefficient is introduced that is shown to be resistant to the effects of outliers. Perhaps even more importantly, a methodology to consider uncertainty in the measured correlation is proposed. Several important implementation details are discussed.

Selected natural fracture networks are presented and analysed in Chapter 4. In one of the major contributions of this chapter, it is demonstrated that these natural fracture networks cannot be modelled using the traditional Poisson process-based simulation techniques in a geologically realistic manner. This finding leads into the next major contribution of this thesis, whereby a new DFN simulation algorithm is proposed. The proposed algorithm is capable of simulating DFNs that match target histograms of fracture spacing, deviation in local orientation of fractures, fracture length, fracture intensity and the number of fracture intersections. Many important implementation details are discussed and case studies on how to implement the ideas are also presented.

1.5 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 introduces a new way to calculate a robust correlation and assess its uncertainty within the context of geological modelling. Chapter 3 discusses implementation details and examples for the proposed robust correlation coefficient and its uncertainty. Chapter 4 looks at a few selected natural fracture networks and discusses whether or not they could be modelled by traditional DFN simulation algorithms. A new fracture simulation algorithm is proposed. Chapter 5 discusses numerous important implementation details surrounding the proposed fracture simulation algorithm. Chapter 6 presents a case study. The fracture modelling methodologies and algorithms are applied to the Teapot Dome data set. Chapter 7 provides further discussion of the ideas presented in this thesis along with conclusions and limitations of this work. Areas for future research are also discussed.

Chapter 2

On a Robust Correlation Coefficient Considering its Uncertainty

This chapter is organized as follows. Section 2.1 introduces the correlation coefficient. Section 2.2 discusses Pearson's and Spearman's correlation coefficients and discusses their sensitivity to outliers. A discussion on outlier data follows and three robust estimates of correlation are reviewed. A new robust correlation coefficient is proposed in Section 2.3. Section 2.4 discusses the uncertainty in the correlation coefficient, which is accessed with its sampling distribution. The distribution of r requires the number of independent data, so a methodology for calculating an effective number of independent data from spatially dependent data is introduced.

2.1 Introduction to the Correlation Coefficient

The relationship between bivariate data, x and y, is frequently summarized by the correlation coefficient, which indicates the level of dependence between two variables. For example, in petroleum reservoirs there is often some correlation between porosity and permeability of the rock matrix.

The sign of the correlation coefficient is positive if the variables are directly related and negative if they are inversely related. The closeness to +1 or -1 measures the closeness to a linear relationship. In some instances a few outliers significantly decrease an otherwise high correlation. The traditional Pearson correlation coefficient is known to be highly affected by outlier data (Abdullah, 1990; Isaaks and Srivastava, 1989; Kim and Fessler, 2004; Shevlyakov, 1997). The Spearman rank correlation coefficient is considered to be more resistant to outliers although it is also adversely affected by outlier data.

Methods for detecting outlier data have been suggested (Barnett and Lewis, 1994; Davies and Gather, 1993; Johnson and Wichern, 2007; Penny and Jolliffe, 2001). Outliers could be trimmed from the data and the correlation of the remaining points can be calculated. However, in some cases, the outlier data may be reliable data and should not be excluded (Gideon and Hollister, 1987), especially when the sample size is small. However, the influence of such data should not be inordinately large.

Often, correlations are estimated from a small number of observations. When the sample size is small, the uncertainty about the value of the true correlation can be very large, particularly when the estimated correlation is low (Kalkomey, 1997). It is useful to quantify the uncertainty in the correlation coefficient to assess its significance and to perform sensitivity studies.

Many statistical and geostatistical models and techniques rely on the correlation between different data variables. This research establishes a procedure to calculate a robust correlation and quantifies the uncertainty in the correlation coefficient through its sampling distribution of the correlation coefficient.

The correlation coefficient is particularly important in cases with sparse primary data and exhaustive secondary data such as offshore petroleum well data and seismic data. In this case, there may be only five to eight wells that have been drilled for production potential and not statistical representivity. Each of these wells is expensive and important. The final geological models will be highly dependent on the correlation coefficient established by simple spreadsheet calculations. Making this correlation robust and understanding its uncertainty has a large practical impact.

2.2 Measures of Correlation

2.2.1 Pearson's Correlation Coefficient

Let $(x_1, y_1), ..., (x_n, y_n)$ be n observations from a bivariate normal distribution with parameters $(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho)$, where μ_x and σ_x^2 are the mean and variance of x, μ_y and σ_y^2 are the mean and variance of y, and ρ is the correlation coefficient between x and y given by $\rho = \beta\sigma_x/\sigma_y$ where β is the slope parameter of regression of y on x. The sample correlation coefficient commonly used for estimating ρ is the Pearson's product-moment correlation coefficient defined by (Pearson, 1920; Rodgers and Nicewander, 1988):

$$r_p = \frac{\sum\limits_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum\limits_{i=1}^{n} (x_i - \bar{x})^2 \sum\limits_{i=1}^{n} (y_i - \bar{y})^2\right]^{1/2}} \tag{2.1}$$

One problem with using Pearson's product-moment correlation coefficient is that the sample means for x and y are sensitive to outlier data. As a result, the correlation estimate r_p is also sensitive to outliers in either x, y, or both variables (Abdullah, 1990; Kim and Fessler, 2004). Even a few outliers can degrade the sample correlation coefficient.

2.2.2 Spearman's Rank Correlation Coefficient

As an alternative to Pearson's correlation coefficient, the non-parametric Spearman's rank correlation coefficient, r_s, can be calculated as follows:

$$r_s = 1 - \frac{6\left[\sum\limits_{i=1}^{n} (r_{y_i} - r_{x_i})^2\right]^2}{n(n^2 - 1)} \tag{2.2}$$

where r_{x_i} and r_{y_i} are the ranks of x_i and y_i, respectively. Spearman's correlation coefficient does not require the assumption of a linear relationship between the variables and is generally more resistant to outliers than Pearson's coefficient. However, as is shown later in this thesis, Spearman's rank correlation coefficient is still quite sensitive to outliers, particularly in the presence of sparse data.

2.2.3 Outlier Data

Outliers can be loosely defined as observations which appear to deviate markedly from the other members of the sample (Grubbs, 1969). Hawkins (1980) defines an outlier as an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism. There is no mathematical definition for what constitutes an outlier (Davies and Gather, 1993) and determining which data (if any) are outliers remains subjective. Data quality control and checking should identify erroneous data for removal. The remaining issue is multivariate observations that influence the calculated statistics.

Outliers can occur in two main ways. They may occur due to random variability in the data. In this case, outliers would normally be generated from a heavily-tailed distribution. The second way for outliers to occur is when the data arise from two different underlying distributions. The "good" data comes from one distribution and the "bad" or "contaminated" data comes from another distribution. In this case, the contaminated data could be due to experimental or measurement error or any number of other ways.

If the data come from a heavily-tailed distribution, the outliers are valid and should not be discarded. In this case those observations should be kept and used. When the outliers occur from another distribution, one would hope to be able to identify and discard those values or use statistical methods that are robust to outliers. Datasets with outliers from another distribution may be caused by mixing data from different geological structures or processes.

Outlier detection has been widely discussed in the literature. Barnett and Lewis (1994), in particular, provide extensive reviews on this topic giving over 100 discordancy tests for a number of distributions. Despite the number of options for detecting outliers, there is no guarantee of finding any because there may not be a test developed for a particular combination, or the data does not follow any standard distribution.

Outlier detection in the bivariate or multivariate case can be even more challenging than in the univariate case. If the bivariate data set is large and highly correlated, detecting outliers may be relatively easy. However, if the data set is small (say less than 20 paired data values) and the correlation is low to moderate (say less than 0.5), it may be difficult to tell whether or not suspicious data points are outliers or an example of lack of correlation between the two variables.

2.2.4 Robust Estimates of Correlation

The idea behind robust estimation of means or covariances (and hence correlation) is to reduce the effect of outlier samples either by weighting or removing them altogether (Campbell, 1980; Kim and Fessler, 2004; Rousseeuw and Zomeren, 1990; Titterington, 1978).

One of the most popular robust methods for estimating correlation (and regression coefficients) is the least median of square (LMS) estimation (Rousseeuw, 1984). The LMS regression coefficients minimize the median of the squared residuals. One of the big advantages of the LMS estimators is their noted 50% break-down point, which means that LMS regression can give reliable results up to the point where 50% of the data are outliers. The LMS algorithm is similar to the bootstrap in that it proceeds by repeatedly drawing subsamples of p different observations from the data set. For each subsample, $J = \{i_1, ..., i_p\}$, a regression line is found for the p points. Each regression line is viewed as a trial estimate and denoted θ_J. For trial, θ_J, the residuals between the regression line and the full data set are calculated. The LMS objective function is defined by the median of the residuals:

$$\underset{i=1,\ldots,n}{\text{med}} \; (y_i - \mathbf{x}_i\theta_J)^2 = \text{median}\{(\mathrm{y}_1 - \mathrm{x}_1\theta_1)^2, (\mathrm{y}_2 - \mathrm{x}_2\theta_2)^2, ..., (\mathrm{y}_\mathrm{n} - \mathrm{x}_\mathrm{n}\theta_\mathrm{n})^2\} \tag{2.3}$$

The trial estimate which gives the minimal median of the squared residuals gives the LMS coefficients and the correlation.

Shevlyakov (1997) introduced a robust correlation coefficient that utilizes the Hampel medians of absolute deviations to get the median correlation coefficient.

$$r_{med} = \frac{\text{med}^2 |u| - \text{med}^2 |v|}{\text{med}^2 |u| + \text{med}^2 |v|} \tag{2.4}$$

$$
\begin{aligned}
u_i &= \frac{x_i - \operatorname{med} x}{\operatorname{med}|x_i - \operatorname{med} x|} + \frac{y_i - \operatorname{med} y}{\operatorname{med}|y_i - \operatorname{med} y|}, \quad i = 1, ..., n \\
v_i &= \frac{x_i - \operatorname{med} x}{\operatorname{med}|x_i - \operatorname{med} x|} - \frac{y_i - \operatorname{med} y}{\operatorname{med}|y_i - \operatorname{med} y|}, \quad i = 1, ..., n
\end{aligned} \tag{2.5}
$$

Gideon and Hollister (1987) approach robust correlation from another perspective by introducing a robust rank correlation coefficient based on the principle of maximum deviations.

$$r_g = (d(\boldsymbol{\varepsilon} \circ \mathbf{p} - d(\mathbf{p}))/[N/2] \tag{2.6}$$

Where $\circ$ is a group operation that is a composition of mappings $\boldsymbol{\varepsilon} \circ \mathbf{p} = (N + 1 - p_1, ..., N + 1 - p_N)$, and $\mathbf{p} = \mathbf{p}(x, y)$ is the permutation determined by the sample and ε is the reverse permutation.

$$d_i(\mathbf{p}) = \sum_{j=1}^{N} I\left(r(x_j) \leq i < r(y_i)\right) \tag{2.7}$$

Where $r(x_j)$ and $r(y_i)$ are the ranks of x and y, respectively.

$$d_i(\boldsymbol{\varepsilon} \circ \mathbf{p}) = \sum_{j=1}^{i} I(i < N + 1 - p_j) \tag{2.8}$$

Another method for calculating a robust correlation coefficient involves calculating an ellipsoid and trimming any data that does not fall within the ellipsoid. This technique works best for data that follow a normal distribution and other swarms of data that are elliptical (Titterington, 1978).

Despite these methods aimed at calculating robust correlation coefficients, there is room for improvement. The key idea developed below is to isolate the influence of each individual data pair (and sets of data pairs) and to ensure that the correlation coefficient is robust, yet fairly considers all data. The performance of the proposed correlation coefficient estimator was checked by a simulation study.

2.3 Correlation in Sparse Datasets in the Presence of Outliers

Figure 2.1 shows a scatter plot of the bivariate relationship between two variables, x and y. The scatter plot shows what appears to be a strong direct correlation between

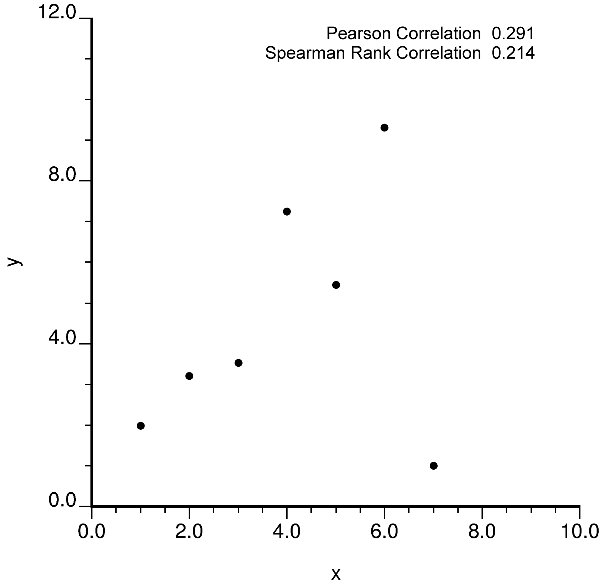


Figure 2.1: An example dataset with one potential outlier data point. The potential outlier appears to be negatively affecting what would otherwise be a strong correlation between x and y. Note that, in this case, the rank correlation coefficient appears to be more strongly affected by the outlier than Pearson's correlation coefficient.

the two variables marred by one potential outlier data point at a location of (7, 1). The Pearson and Spearman correlation coefficients for the data shown in Figure 2.1 are 0.291 and 0.214, respectively. Note that, while the Spearman correlation coefficient is usually more resistant to the effects of outliers, in this case it is more strongly affected by the potential outlier data point.

Of course when a dataset such as the one shown in Figure 2.1 is observed, it would be natural to think that the point at (7,1) is an outlier. It may be a sample that belongs to another statistical population or perhaps there was an error in measurement.

If one could be reasonably sure that the point at (7, 1) is an outlier, it could simply be removed from the dataset. In this case, the Pearson and Spearman correlations would increase to 0.910 and 0.943, respectively. However, suppose the data has been carefully examined and there is no reason to believe that the point at (7,1) belongs to another statistical population. In this case, the data should be considered in the calculations, but its importance should not be unreasonably large.

Table 2.1: Resulting correlations from a leave-one-out test (actual correlation is 0.29)

Coordinates of left out point (x,y)	Resulting correlation	Weight (w_i)
(1.00, 1.98)	0.081	0.085
(2.00, 3.20)	0.235	0.010
(3.00, 3.53)	0.269	0.002
(4.00, 7.25)	0.318	0.003
(5.00, 5.44)	0.272	0.002
(6.00, 9.31)	0.002	0.140
(7.00, 1.00)	0.910	0.468
Updated correlation from leave-one-out test = 0.615		

2.3.1 A Weighted Average Correlation From a Leave-One-Out-Test

In order to arrive at a more robust correlation coefficient, first consider a "leave-one-out test" (LOOT), whereby a data point is removed from the dataset and the correlation is recalculated. This procedure can be repeated n times for a dataset with n points, leaving a different data point out each time. The result is n calculated Pearson correlation coefficients. A LOOT was conducted for the data shown in Figure 2.1 and the results are shown in Table 2.1. For the first six leave-one-out tests, the resulting correlations are very low and unrepresentative of the obvious correlation in the data. However, the last test results in a correlation of 0.910.

The proposed robust correlation coefficient is based upon the idea of a weighted average of the correlations calculated in the LOOT. The idea is to weight the correlations according to their difference from the actual correlation as follows:

$$w_i = |r_{Actual} - r_{i,LOOT}|^{\alpha} \tag{2.9}$$

where:

- w_i is the resulting weight assigned to each correlation calculated in the leave one out test;
- r_{Actual} is the Pearson's correlation coefficient calculated using all of the original data;
- $r_{i,LOOT}$ is the i^{th} Pearson's correlation coefficient calculated from leaving out the i^{th} data point in the leave-one-out test;

- α is a weighting exponent and is a function of the number of data ($\alpha = 1+n/12$). α is restricted to a maximum of 15 due to computational limitations and the observation that beyond a certain point, a larger exponent is unnecessary.

The weighting exponent, α, is directly related to the sample size. As the sample size increases, the outlier data point effectively gains more influence on the calculated correlations in the LOOT since there are more combinations that use the outlier and only a few that do not. Thus, a larger exponent gives a larger weight (relative to the other weights) to the correlation that does not use the outlier data point.

In the above example the weights from the LOOT are shown in 2.1. The correlation obtained from removing the point at (7, 1) is the most different from the actual correlation of 0.291 and thus receives the most weight. This makes sense since it is desirable to somehow minimize the impact of this suspicious data point. Then, the more robust correlation coefficient calculated from the LOOT for sparse datasets is defined as:

$$r_{Robust,LOOT} = \frac{\sum\limits_{i=1}^{n} w_i r_{i,LOOT}}{\sum\limits_{i=1}^{n} w_i} \tag{2.10}$$

The updated correlation coefficient is essentially a weighted average of the correlations calculated in the LOOT, where the weights are defined in Equation 2.9. The weighting scheme in Equation 2.9 assigns the greatest weights to the correlations that are the most different from the actual Pearson's correlation coefficient. The idea is that the data points that have the biggest impact on the correlation are the ones that are most likely to be outliers. For the data shown in Figure 2.1, the LOOT correlation weights and updated correlation coefficient are as follows in Table 2.1. As is shown in the table, the updated correlation coefficient from the LOOT is 0.615, which seems reasonable given that it is undesirable to exclude the potential outlier.

2.3.2 A Weighted Average Correlation From a Leave-X-Out Test

The LOOT and weighted average correlation is effective for the case where there is one potential outlier data point. Of course, the idea can be extended to account for multiple potential outlier data points by considering a "Leave-X-Out Test" (LXOT) where X varies from 1 to $n-3$ (correlations cannot be calculated with 1 data and correlations calculated using 2 data have little value). This yields $n-3$ weighted average correlations from each of the LXOTs. Then, each of the weighted average correlations can be weighted again in a similar manner as follows:

$$w_{X,LXOT} = |r_{Actual} - r_{X,LXOT}|^{\alpha} \tag{2.11}$$

Where:

- r_{Actual} is the original data correlation;
- $r_{X,LXOT}$ is the updated correlation calculated in each "leave-X-out-test";
- $w_{X,LXOT}$ are the weights calculated for each updated correlation from the leave "x-data" out test;
- α is the same weighting exponent as in Equation 2.9 (i.e. $\alpha = 1 + n/12$).

The weighting exponent in Equation 2.11 works similarly to that of Equation 2.9. We wish to access the correlation from the LXOTs that are the most different from the original Pearson correlation. A larger exponent for larger sample sizes effectively gives more weight to the correlations from the LXOTs that are the most different from the original Pearson correlation.

Then a single robust correlation is calculated as follows:

$$r_{robust-proposed} = \frac{\sum\limits_{X=1}^{\phi} w_{X,LXOT} r_{X,LXOT}}{\sum\limits_{X=1}^{\phi} w_{X,LXOT}} \tag{2.12}$$

where:

$$\phi = 0.8n - 3 \tag{2.13}$$

ϕ is rounded up to the nearest integer. Equation 2.13 allows for a reasonable maximum number of data to leave out.

As the number of data left-out increases the updated correlations calculated using only small amounts of data tend to be unreliable. For example, say there are 30 data and a L27OT is conducted (where varying combinations of 27 data are left out and the remaining 3 data are used). The resulting correlations from the L27OT would likely be very erratic depending on which data are selected. Even for highly positively correlated data there would be some subsets of 3 data that would form a strong negative correlation. Since the correlations are weighted by their difference to the actual correlation, those subsets would receive a lot of weight. Thus, it is suggested that constraining X from 1 to ϕ, as in Equation 2.13. Thus, in a case with 100 data points, the proposed correlation considers LXOTs where X ranges from 1 to 77. There is no need to consider leaving out very large subsets of data anyway. If there are 100 data points, leaving out a maximum of 77 (a L77OT) will already consider subsets with no outliers since there are normally far less than 77/100 outliers in the data. If there were 77 data that appeared to be from one distribution and 33 from another, one would normally call the 33 data outliers.

2.4 The Distribution of r

It is also necessary to examine the uncertainty in the correlation coefficient. Since datasets usually have limited sample information, the true underlying correlation is rarely known. When the sample size is small, the uncertainty in the correlation coefficient can be very large, particularly when the measured correlation is low (Kalkomey, 1997).

The distribution of r (the sample correlation coefficient) as given in Johnson et al. (1995) is as follows:

$$p_R(r) = \frac{(1-\rho^2)^{(n-1)/2}(1-r^2)^{(n-4)/2}}{\sqrt{\pi}\Gamma(\frac{1}{2}(n-1))\Gamma(\frac{1}{2}n-1)} \times \sum_{j=0}^{\infty} \frac{[\Gamma(\frac{1}{2}(n-1+j))]^2}{j!}(2\rho r)^j \qquad (2.14)$$

where $-1 \leq r \leq 1$. In the equation, ρ is the estimated correlation, n is the number of *independent* data points and Γ is the gamma function. Note that Equation 2.14 also assumes that (X_i, Y_i) and (X_j, Y_j) are mutually independent if $i \neq j$.

2.4.1 Calculating the Number of Independent Data

Equation 2.14 requires the number of independent data points. However, earth sciences data are rarely independent and are often spatially related. We can, however, calculate an effective number of independent data.

Consider a number of observations X_i, where $i = 1, \ldots, n$. The variance of the mean is given by:

$$Var\{\bar{x}\} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} \tag{2.15}$$

where C_{ij} is the data covariance. The covariance of the data can be calculated from the variogram and the data variance:

$$C_{ij} = \sigma^2_{data} - \gamma_{ij} \tag{2.16}$$

But, the variance of the mean can also be calculated by (Edwards, 2006):

$$Var\{\bar{x}\} = \frac{\sigma^2_{data}}{N_{independent}} \tag{2.17}$$

Where $N_{independent}$ is the number of independent data. Therefore:

$$N_{independent} = \frac{\sigma^2_{data}}{Var\{\bar{x}\}} = \frac{\sigma^2_{data}}{\frac{1}{n^2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} C_{ij}} = \frac{\sigma^2_{data}}{\frac{1}{n^2}\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\left(\sigma^2_{data} - \gamma_{ij}\right)} \tag{2.18}$$

Simplifying, we have:

$$N_{independent} = \frac{n^2\sigma^2_{data}}{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}\left(\sigma^2_{data} - \gamma_{ij}\right)} \tag{2.19}$$

Thus, the effective number of independent data can be calculated using only the number of data and the variogram. When the correlation between two variables is being considered, the variogram with the longest range should be used since it will yield a lower effective number of independent data, which results in a larger uncertainty in ρ.

2.5 Remarks

The relationship between multiple variables in geostatistics is frequently estimated using the correlation coefficient. Reliable incorporation of secondary data relies on obtaining a robust measure of correlation between primary and secondary variables. However, where there are a small number of samples, the uncertainty in the correlation coefficient can be very large (even when a robust correlation coefficient is calculated), and should be considered in successive geostatistical calculations.

If a scatter plot of the data indicates the possibility for outlier data, the first step is to examine the data for any potential errors or inaccuracies. If no errors in the data are found and the sample is small, the engineer or geologist may not wish to eliminate the suspicious data points from the dataset.

This chapter has introduced a methodology for estimating a robust correlation coefficient (Equation 2.12) and estimating its uncertainty (Equation 2.14) using the sampling distribution of the correlation coefficient. The number of independent data is required for the sampling distribution. Although most geological datasets are spatially correlated, the number of independent data can be calculated with Equation 2.19. The next chapter discusses several important implementation aspects and compares the proposed robust correlation coefficient with others.

Chapter 3

Practical Aspects of Calculating a Robust Correlation and Calculating its Uncertainty

The previous chapter proposed a new way to calculate a robust correlation coefficient and its uncertainty. This chapter presents the implementation and practical calculation details of the proposed robust correlation coefficient and its uncertainty. `FORTRAN` computer programs are introduced to perform the correlation calculations. Finally, the proposed robust correlation coefficient is compared to other robust correlation coefficients.

3.1 Computer Codes

A suite of three computer programs were created to estimate a robust correlation coefficient and its associated uncertainty.

A `FORTRAN` program called `ROBUSTCORRCO` automatically calculates the correlation coefficients for each LXOT as well as an overall robust correlation coefficient, as in Equation 2.12. In cases where there are more than approximately 20 data points, the time to calculate the number of combinations of data in the LXOT becomes prohibitively large. As a result, a specified number (say 10,000) of data combinations are randomly sampled rather than calculating the correlation for every possible data combination. `ROBUSTCORRCO` also calculates the two traditional correlation coefficients as well as Shevlyakov's r_{med}, Gideon and Hollister's r_g, and Rousseeuw's r_{LMS} for comparison purposes.

A `FORTRAN` program called `NIND` automatically calculates the effective number of independent data based on Equation 2.19 and the input variogram model.

A `FORTRAN` program called, `SAMP_DIST_CORR`, calculates the sampling distribution for the correlation coefficient. The program uses Equation 2.14 with the measured data correlation and the effective number of independent data as inputs.

3.1.1 Practical Considerations

Although the summation in Equation 2.14 is to infinity, it tends to converge rapidly except where the measured correlation is quite high (i.e. $\rho > 0.9$). Thus, an upper summation limit and a tolerance parameter are specified inputs into the `SAMP_DIST_CORR` program. The program calculates the percentage of instances where the summation parameter does not converge to a value smaller than the specified tolerance parameter. If the percentage of values not converging is too high, the summation parameter can be increased (or the tolerance can be increased).

3.2 Breakdown Properties of the Proposed Robust Correlation Coefficient

A simulation study, similar to the one presented in Abdullah (1990), illustrates the breakdown properties of the proposed robust correlation coefficient 2.12 compared to the traditional Pearson and Spearman correlation coefficients, as well as the three robust correlation coefficients, r_{med}, r_g and r_{LMS}, proposed by Shevlyakov (1997), Gideon and Hollister (1987) and Rousseeuw (1984), respectively.

First, 100 "good" observations are generated according to the linear relation $y_i = 2 + x_i + u_i$ where x_i is drawn randomly from a normal distribution with a mean of 5.0 and a variance of 1.0. u_i is drawn from a normal distribution with a mean of 0 and a standard deviation of 0.2. The results were as follows: $r_{Pearson} = 0.974$, $r_{Spearman} = 0.969$ and $r_{robust-proposed} = 0.906$. Note that the proposed correlation ($r_{robust-proposed}$) is slightly lower than the Pearson and Spearman coefficients. The original Pearson correlation is quite high (0.974), so when the proposed algorithm leaves out data near the tips of the bivariate distribution, a slightly lower correlation is measured in the remaining data, which has the biggest impact on the proposed robust correlation. However, if the measured Pearson correlation was lower (say 0.5), this effect would be less pronounced.

Next, the data was slowly contaminated. In increments of 10 data points, the "good" data was replaced with "bad" data points. The contaminated data points were generated according to the linear relation where x_i is uniformly distributed on [5, 10] and y_i is drawn from a normal distribution with a mean of 2 and a standard deviation of 0.2.

This was repeated until only 50 "good" observations remained. Figure 3.1 shows the comparison of the proposed robust correlation coefficient against the traditional Pearson and Spearman correlation coefficients as well as three other robust correlation coefficients and serves to highlight the point at which the correlation coefficients begin to breakdown. In this study, Pearson's correlation coefficient breaks down with less than 10% contamination. Spearman's is more robust, as expected, but is still significantly affected by just 10% contamination. Gideon and Hollister's r_g fares only slightly better than Spearman's correlation coefficient and its measured correlation with no contamination is much lower than the others. Shevlyakov's r_{med} exhibits reasonable resistance to data contamination until about 20% contamination, but by 30% contamination the correlation drops substantially. Rousseeuw's least median of square correlation, r_{LMS}, is known to have a 50% breakdown point, as is shown in the figure. This is one of the main advantages of LMS regression. The proposed robust correlation is significantly better than all except Rousseeuw's LMS coefficient.

3.3 Examples

Figure 3.2 shows a porosity versus log permeability dataset with twelve paired points. Each point is labeled with an arbitrary number for reference purposes. Figure 3.3 shows the location maps for the porosity and log permeability values. The left side of the circles indicate porosity (in %) and the right side of the circles indicates Log_{10} permeability (in mD). The Pearson and Spearman correlation coefficients between the

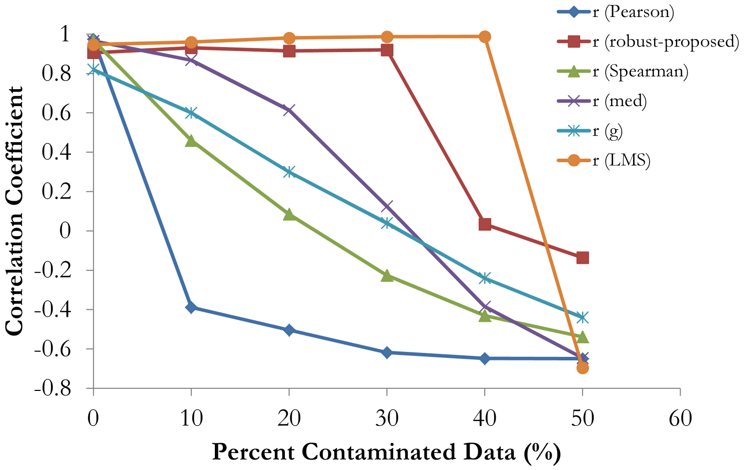


Figure 3.1: Simulation study comparing the effect of contaminated data on the Pearson, Spearman and proposed robust correlation coefficients and three other robust correlation coefficients.

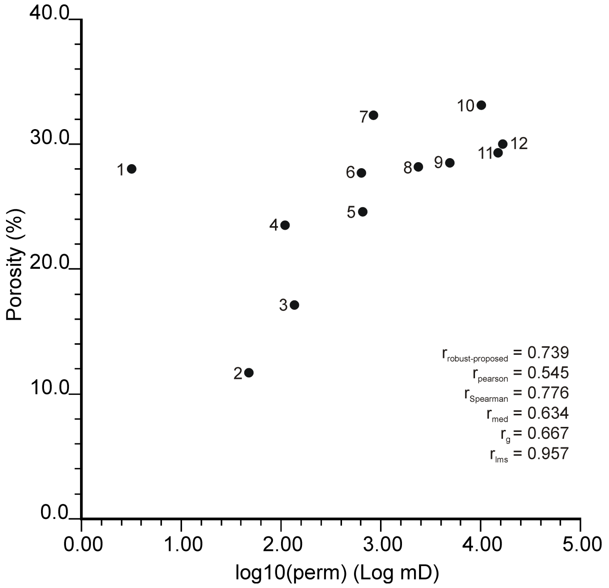


Figure 3.2: A sparse synthetic dataset.

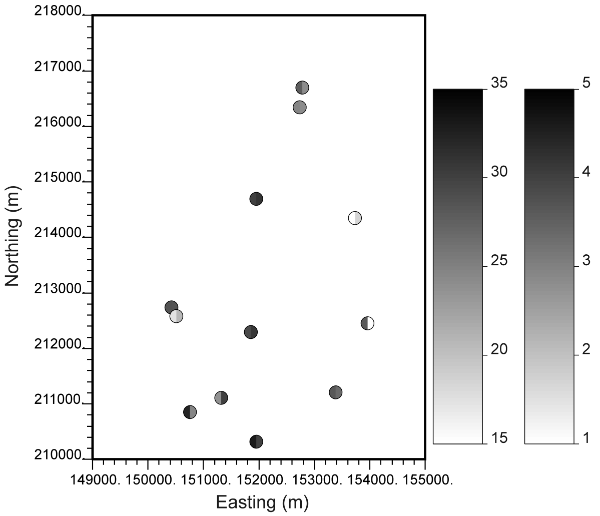


Figure 3.3: Location map of twelve samples. The left scale and left side of the circles indicate porosity (in %). The right scale and right side of the circles represent Log_{10} permeability (in mD).

porosity and log permeability data are 0.545 and 0.776, respectively. Since Pearson's correlation is lower than Spearman's rank correlation coefficient, Pearson's correlation may be affected by outlier data. Visual inspection of the scatter plot in Figure 3.2 confirms that data point number one, in particular, and to a lesser extent two, appear to be "suspicious" or outliers. In this example, there are no known errors in the measurements. The program `ROBUSTCORRCO` calculates a robust correlation coefficient of 0.739, which agrees with the Spearman rank coefficient. The program also calculates three other robust correlations, which are also noted on Figure 3.2. In this case Shevlyakov's r_{med} and Gideon and Hollister's r_g are slightly lower than the proposed robust correlation and Spearman's rank correlation. However, here the LMS correlation coefficient is 0.957, which seems too high based on visual inspection of the data and is much higher than any of the other correlation coefficients.

With knowledge of the estimated and proposed robust correlation coefficients, the uncertainty in the correlation coefficient can be calculated using the sampling distribution for the correlation coefficient. First, however, the number of independent data points must be calculated. The program `NIND` can be used to calculate the effective number of independent data points. The data file and a variogram model are the only two inputs into the `NIND` program. In this case, a single-structure spherical variogram

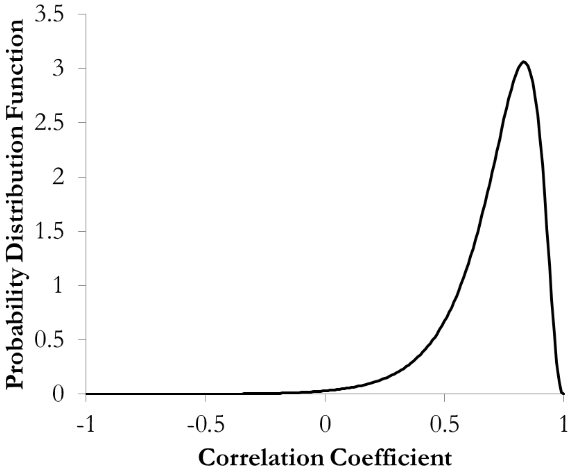


Figure 3.4: The sampling distribution for the correlation coefficient for the synthetic core data shown in 3.2.

with a range of 4000 units in the horizontal plane and 10 units in the vertical direction was assumed since there is not enough data to calculate a reliable experimental variogram. Based on the data configuration, the number of data and the assumed variogram, the effective number of independent data calculated by `NIND` is 11.

The program `SAMP_DIST_CORR` is used to calculate the sampling distribution for the correlation coefficient. The robust correlation (0.739) and the number of independent data ($N_{ind} = 11$) are input into the program. The output is a probability density sampling distribution for the correlation coefficient, which is shown in Figure 3.4. Note that the P50 for this distribution is approximately 0.76, which is different than the mean due to the asymmetric nature of the sampling distribution. The 10^{th} and 90^{th} percentile (the P10 and P90) correlation values are approximately 0.11 and 0.90, respectively.

Note that if the measured correlation was lower, or if there were fewer data, the sampling distribution for the correlation coefficient would be even wider. For example, Figure 3.5 shows the sampling distribution for the correlation coefficient for a measured correlation of 0.3 and 8 independent data points. The uncertainty in the correlation is much wider in this case and the P10 and P90 correlation values are -0.34 and 0.78, respectively.

For one final example, consider the data from an offshore reservoir shown in Figure 3.6. The figure shows six paired points on a scatter plot between a seismic attribute

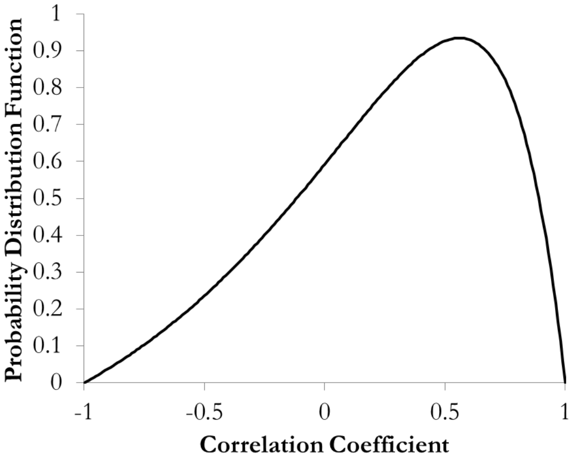


Figure 3.5: The sampling distribution for the correlation coefficient for a measured correlation of 0.3 and six independent data points.

and porosity. The data points have been labelled with arbitrary numbers for reference purposes. Pearson's correlation is -0.471 and Spearman's rank correlation is -0.771. Upon examination, data points one and six appear to be outliers, or at least suspicious. Here, the program `ROBUSTCORRCO` calculates a robust correlation coefficient of -0.533, which is in good agreement with the traditional Pearson correlation coefficient. This makes sense when the results are examined in more detail. When point number one is left out of the calculation in a LOOT, the correlation between points two to six is -0.087. However, when point number six is left out of the calculation, the correlation of the remaining points is -0.927. Elimination of any of the other points makes little difference to the resulting correlation. Thus, the effect of point one and six roughly offset each other.

The three robust correlations are also indicated on Figure 3.6. As is shown, the robust correlation coefficients are $r_{med} =$ -0.351, $r_g =$ -0.667 and $r_{LMS} =$ -0.972. Just as in the last example the LMS correlation tends toward the extreme end of the correlation spectrum.

Figure 3.7 shows the sampling distribution for the robust correlation coefficient for the offshore reservoir data. In this case, there are no spatial locations for the data so it is assumed that the data are independent of each other. The sampling distribution shows P10/P50/P90 correlation values of approximately -0.89/-0.60/0.00, respectively.

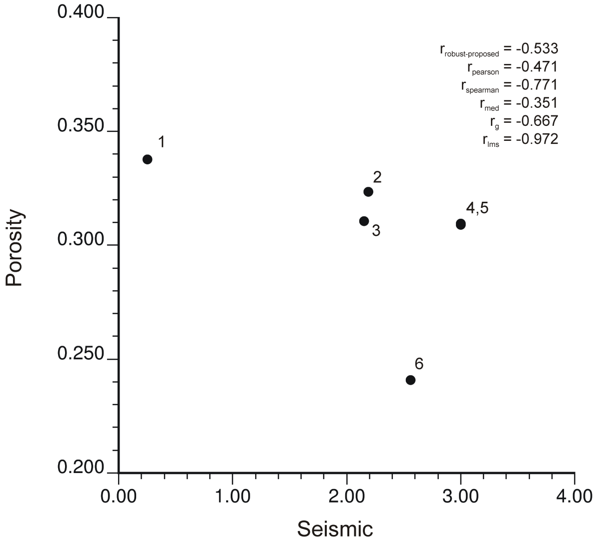


Figure 3.6: A seismic attribute versus porosity from an offshore reservoir. Although only five points are visible, there are actually six points since there are two points very close together near (3, 0.31).

Thus, according to its distribution, there is a 10% chance that the correlation is greater than zero, based upon the number of data and the calculated robust correlation.

Simulation Study

In an effort to compare the proposed robust correlation coefficient to the other robust correlation coefficients and to help explain the extreme r_{LMS} values, a few small simulation studies were performed. In the first simulation study, 100 realizations of 10 data points (x, y) are generated by drawing x and y values randomly and independently from a uniform distribution between 0 and 10. Correlation coefficients are calculated for each realization of 10 data. Since the x and y values are drawn randomly and independently from uniform distributions, the average correlation is expected to be 0.0. The results of the study are shown in Table 3.1. For each correlation coefficient, the average was very close to 0.0, as expected. More interesting is the standard deviation of correlation of the 100 realizations. The $\sigma_{Pearson} = \sigma_{Spearman} = 0.34$ while $\sigma_{robust-proposed} = 0.29$ and is similar to $\sigma_g = 0.25$. The lower standard deviation for $\sigma_{robust-proposed}$ and σ_g makes sense since they should be less affected by outliers that give spurious correlation to Pearson and Spearman's coefficients for some realizations.

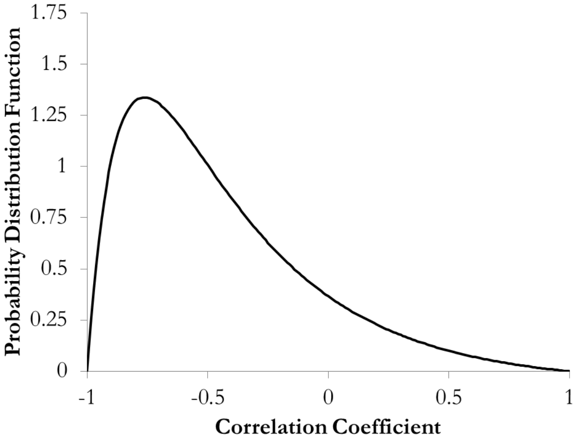


Figure 3.7: The sampling distribution for the correlation coefficient for the data in Figure 3.6.

Interestingly, $\sigma_{LMS} = 0.711$, which is much larger than for any other correlation coefficient. Figure 3.8 provides additional insight showing the relative frequency histograms of $r_{pearson}$, $r_{robust-proposed}$ and r_{LMS}. The histograms of $r_{Pearson}$ and $r_{robust-proposed}$ are roughly symmetric and centered around 0.0, as expected. However, the histogram for r_{LMS} shows a distinct tendency towards values near -1 and 1. Figure 3.9 shows similar relative frequency histograms except that the number of data points per realization was increased from 10 to 50. When the number of data per realization is increased, the standard deviation of the correlations decreases the chance of generating a data set with spurious correlation. However, even with 50 data points generated from two independent uniform distributions, the range of correlation calculated by the LMS algorithm remains very wide. The standard deviation for $r_{LMS} = 0.509$ with minimum and maximum correlations of -0.98 and 0.99.

For another simulation study, 100 realizations of 10 data points are generated. This time the x-values were drawn from a normal distribution with $m_x = 15$ and $\sigma_x = 4$. The y-values are drawn from a normal distribution with a mean conditional to x ($m_{y|x}$), conditional variance ($\sigma^2_{y|x}$) and a target correlation of 0.6 where:

$$m_{y|x} = m_y + \rho\sigma_y \frac{(x - m_x)}{\sigma_x} \tag{3.1}$$

and

Table 3.1: Simulation study results for 100 realizations of 10 data points drawn from independent uniform distributions.

	Correlation coefficient					
	Pearson (r_p)	Spearman (r_s)	Proposed robust ($r_{robust-proposed}$)	Shevlyakov (r_{med})	Gideon & Holister (r_g)	Rousseeuw (r_{LMS})
Average r	-0.040	-0.035	-0.039	-0.026	-0.044	-0.035
St. Dev. (σ)	0.340	0.340	0.289	0.444	0.247	0.711
Minimum r	-0.852	-0.952	-0.792	-0.907	-0.600	-0.991
Maximum r	0.711	0.842	0.666	0.873	0.400	0.989

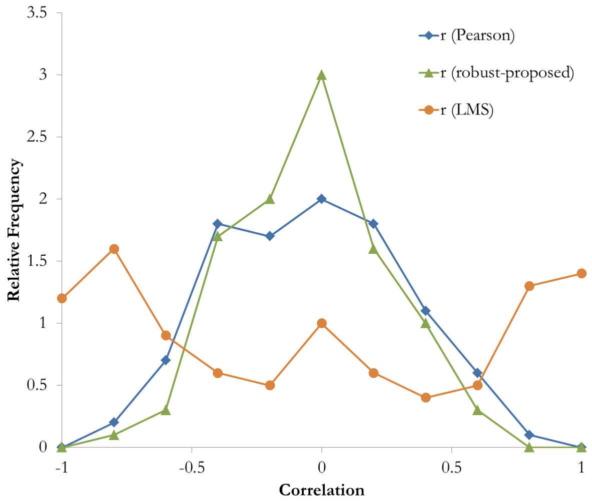


Figure 3.8: Relative frequency histograms of correlation for 100 realizations of 10 data points measured by three correlation coefficients, $r_{Pearson}|$, $r_{robust-proposed}$ and r_{LMS}.

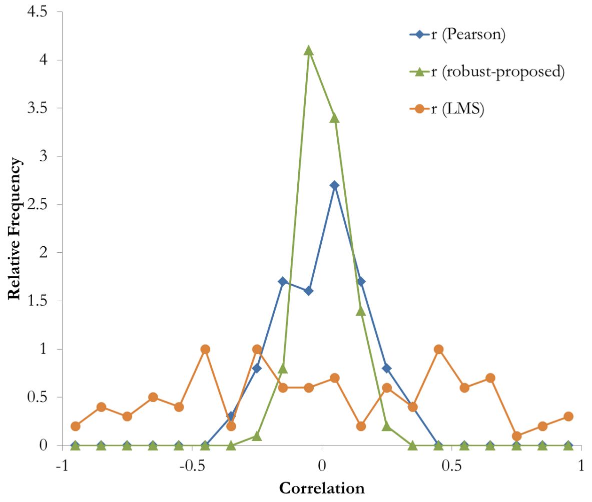


Figure 3.9: Relative frequency histograms of correlation for 100 realizations of 50 data points measured by three correlation coefficients, $r_{Pearson}|$, $r_{robust-proposed}$ and r_{LMS}.

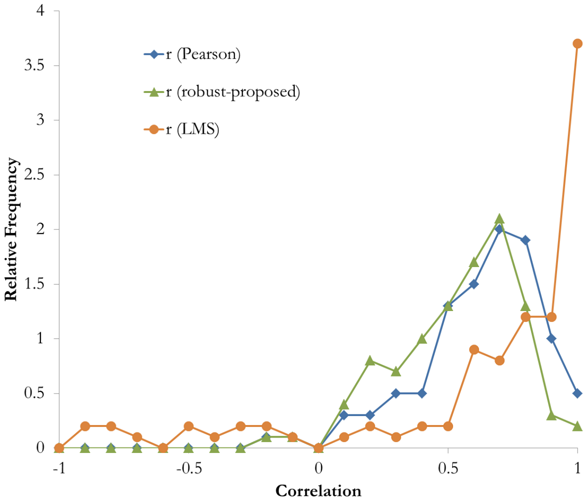


Figure 3.10: Relative frequency histograms of correlation for 100 realizations of 10 data points (with expected correlation = 0.60) measured by three correlation coefficients, $r_{Pearson}|$, $r_{robust-proposed}$ and r_{LMS}.

$$\sigma^2_{y|x} = \sigma^2_y(1-\rho^2) \tag{3.2}$$

Pearson's, the LMS and the proposed robust correlation coefficients are calculated for each of the 100 realizations. The average Pearson correlation is 0.634. The average proposed robust correlation is 0.524 and is lower than the target of 0.6. The average LMS correlation is 0.797, which is considerably higher than the target. The relative frequency histograms are shown in Figure 3.10. The histograms for Pearson's and the proposed robust correlation coefficients are similar although the proposed correlation tends to be slightly lower than Pearson's. However, similar to the previous examples, the LMS correlation has a strong tendency towards the extreme end of the correlation spectrum. In fact, the LMS estimator calculates a correlation greater than 0.9 in 37 of 100 realizations.

3.4 Discussion

Given that there is no formal mathematical definition for an outlier and that the practitioner may or may not wish to exclude outliers depending on their origin and the number of data, it should be no surprise that there are a many ways to calculate a robust correlation coefficient. It appears that no correlation coefficient is perfect in every situation. The traditional correlation coefficients are particularly sensitive to outliers and two of the three robust coefficients (r_{med} and r_g) are slightly more robust, but still sensitive to outliers as shown by the breakdown test. The test also showed the proposed correlation coefficient is quite resistant to outliers, though not as much as the LMS correlation coefficient. However, in examples with lower correlation and fewer data points, the LMS correlation coefficient tends toward the extreme ends of the correlation spectrum. Moreover, for any particular underlying data distribution, the LMS estimator may calculate a correlation anywhere along the spectrum depending on the particular data configuration.

It seems to be a good idea to calculate and compare several correlation coefficients for any particular dataset. If some of the coefficients agree with each other, it may be easier to trust one of those calculated values. On the other hand, visual inspection of a scatter plot of the data should not be overlooked. If the calculated correlation appears to disagree with visual inspection of the scatter plot, perhaps one of the other robust or traditional correlation coefficients makes more sense in that situation.

One example of application of this research is in collocated co-simulation. In collocated co-simulation a Markov-type assumption is made where collocated secondary information is assumed to screen further away data of the same type. This means that the available primary data and a single secondary datum at the estimation location are used in the calculation (Deutsch, 2002). The collocated co-simulation relies on the measured correlation between the primary and secondary data. In some cases such as in off shore oil and gas reservoirs, there may be few wells or samples upon which a correlation may be calculated. In these cases, the uncertainty in the correlation coefficient may be quite large.

It is useful and valuable to use the proposed robust correlation coefficient (encoded in the program `ROBUSTCORRCO`) to obtain a more robust correlation coefficient. Then the effective number of independent data (encoded in the program `NIND`) can be calculated. Finally the program `SAMP_DIST_CORR` can be used to obtain the sampling distribution for the correlation coefficient.

The aim is to arrive at a more robust correlation coefficient which minimizes the effect of outliers and is hopefully more reflective of the true correlation coefficient, which will never be known. By using a more robust correlation as input into the collocated cosimulation, the influence of secondary data can be more correctly scaled in the estimation and simulation of the variable of interest.

The use of the sampling distribution for the correlation coefficient allows one to examine the impact of the secondary data on our estimates. In this example, the geostatistician could calculate three scenarios of collocated co-simulation with the P10, P50 and P90 correlation coefficients to observe the impact of the uncertainty in the correlation on the measured reserves. Or, a Monte Carlo simulation approach could be used to randomly sample the distribution of r as an input into the collocated co-simulation. Given the demonstrated uncertainty in the correlation coefficient, it is not difficult to imagine different scenarios having a major impact on estimated reserves.

3.5 Remarks

The sensitivity of Pearson's correlation coefficient to outliers is well known. An empirical study of the breakdown properties of the traditional and robust correlation coefficients indicates that Spearman's, Shevlyakov's and Gideon and Hollister's correlation coefficients are also significantly affected by outliers although they are more robust than Pearson's coefficient.

A new robust correlation coefficient was proposed, which showed a higher breakdown point than the traditional correlation coefficient and two other robust correlation coefficients. The LMS correlation coefficient has a breakdown point of 50%, which is the maximum possible. However, it was shown that the LMS correlation coefficient may exhibit a tendency towards calculating extreme correlation values. Moreover, even when realizations are drawn from some underlying distribution with a strong positive correlation, the LMS estimator may calculate low or negative correlations.

Care and judgment should be used in selecting a correlation coefficient to represent a dataset. It is fairly easy and quick to use `ROBUSTCORRCO` to calculate each of the robust correlation coefficients in addition to the traditional ones. Then the calculated values can be compared to each other and a scatter plot of the data to arrive at a reasonable value.

Regardless of the calculated value or choice of correlation between two variables, if the dataset is small, the uncertainty in the correlation can be very large. The sampling distribution for the correlation coefficient can be used to quantify its uncertainty based upon the measured correlation value and the number of independent data.

Finally, the calculated uncertainty in the correlation coefficient should be propagated through geological modeling (or any further statistical or geostatistical analysis) to determine its impact on the resulting models. This can be easily achieved by running scenarios with different percentiles of the correlation coefficient or a Monte-Carlo simulation approach.

Chapter 4

Improvements to Fracture Modelling

This chapter begins by describing how current DFN simulation software generates fracture locations using a Poisson process (essentially the Baecher model, as was introduced in Section 1.2.4). The Poisson process is explained here in detail. Next, two natural fracture networks are examined and it is demonstrated that those fracture patterns can not be replicated using the current DFN simulation software. As a result, a new algorithm for simulating DFNs is proposed.

Generating DFNs as models of fractures in a reservoir is becoming more common, as is evidenced by increasing number of articles in the literature on the topic (Basquet et al., 2004; Casciano et al., 2004; Dershowitz et al., 2004; Guaiquirian et al., 2007; Kim and Schechter, 2007; Makel, 2007; Tran et al., 2006; Xu and Dowd, 2010). Currently, there are only a handful of commercially available computer programs capable of simulating DFNs in the context of a petroleum reservoir including (but not necessarily limited to): FracMan (Golder Associates Ltd., 2010), Roxar's RMS (Emerson Process Management (Roxar), 2011), NAPSAC (Hartley, 1998), Prism's REFRACT (Prism Seismic Inc., 2012) and Schlumberger's Petrel (Schlumberger, 2009). Note that, Petrel directly incorporates Golder's FracMan code in their program.

FracMan, RMS, NAPSAC and Petrel all use a Poisson process (the Baecher or enhanced Baecher model) to randomly generate fracture centroid locations. Fracture orientations are drawn independently of location. Fracture density can vary locally, but there is no control on spacing and local orientation of fractures. No information on REFRACT's fracture simulation algorithm was found in the literature, nor on their website. The nearest neighbour and Levy-Lee models (see Section 1.2.4) are also available in FracMan and Petrel. This chapter mainly examines the modelling of fracture networks using the Poisson process-based Baecher model, rather than the Levy-Lee or the nearest neighbour models since: 1) the nearest neighbour model is generally used for a special case of DFN modelling where fracture intensity is expected to vary in relation to proximity of a fault, 2) the Levy-Lee model may give some control over spacing via the fractal dimension and spacing parameters; however, no consideration is given to intersections or local orientation variations, and 3) the Poisson process-based approaches seem to find far more use in practice and in the literature.

4.1 Poisson Processes

In its simplest form, a **Poisson Process** is a stochastic counting process that describes the number of events in a given time interval. For example, physicists use Poisson processes to describe the emission of radioactive particles. The process is described by a counter $N(t)$ that tells us the number of events in the interval $(0, t)$.

The Poisson process has the following properties:

1. $N(0) = 0$;
2. The number of events in disjoint (not overlapping) intervals are independent;
3. The number of events in a given interval depends only on the length of that interval and not on its particular position in time;
4. No events are simultaneous; and,
5. $P(N(t) = n) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}, n = 0, 1, 2, ...$ The number of events in any interval is Poisson distributed with mean λt. λ is called the rate parameter and is the expected number of events occurring per unit of time.

The process is called a **non-homogeneous Poisson process** if the rate parameter may change with time. Here, the rate function is generalized to $\lambda(t)$. Therefore, the expected number of events between time a and b is:

$$\lambda_{a,b} = \int_a^b \lambda(t)dt \tag{4.1}$$

A variation on the Poisson process is the **spatial Poisson process**. In one-dimensional space, the theory is the same except that the index variable is one-dimensional space instead of time. In higher dimensional spaces, the spatial Poisson process is defined by having a number of points $N(A)$ in a region, $A \in V$, given by the Poisson distribution:

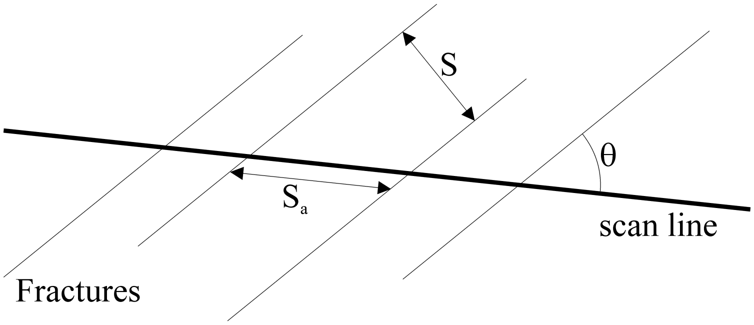
$$P(N(A) = k) = \frac{e^{-\lambda|A|}(\lambda|A|)^k}{k!}, n = 0, 1, 2, ... \tag{4.2}$$

If $(A_i, i \in I)$ is a finite family of pairwise disjoint elements of V, then the random variables $(N(A_i), i \in I)$ are mutually independent. In the spatial case of the Poisson process, the mean value is the number of points within A, which is equal to $\lambda|A|$ and λ is the number of points per unit volume. In the spatial Poisson process, the rate function becomes the **intensity function**. For more information on the spatial Poisson process, the reader is referred to Lantuejoul (2002).

Since Poisson process events (fracture centroids in this case) are independent, centroids may occur very close together (i.e. infinitesimally close together). Using a Poisson process to generate fracture locations leads to negative exponential fracture spacing distributions and fails to account for the clustering of joints (Belfield, 1998). However, some research has indicated that fracture spacing has a log-normally distribution (Narr et al., 2006; Bridges, 1975; Becker and Gross, 1996), indicating that fractures are not distributed independently of each other. Nevertheless, the use of Poisson models to generate fracture locations persists as is evidenced by the state of the commercial DFN simulation software.

4.2 Measuring Fracture Spacing

Geologists and engineers often use scan lines to measure fracture spacing in outcrops. A scan line could also be considered as the centre line through core obtained from diamond drilling. However, the relative orientation of the sampling line or borehole



S_a = apparent fracture spacing
S = true fracture spacing
θ = angle between scanline and fracture
Note: This diagram is scale invariant.

Figure 4.1: Relationship between true and apparent fracture spacing.

compared to the orientation of the fractures introduces a bias in the fracture spacing measurements. The actual fracture spacing is less than the apparent spacing. The bias in fracture spacing is demonstrated in Figure 4.1 and can be corrected using a correction proposed by Terzaghi (1965):

$$S = S_a \sin\theta \tag{4.3}$$

In Equation 4.3, S is the true spacing between discontinuities, S_a is the apparent spacing measured on the scan line (or borehole) and θ is the angle between the sampling line and the orientation of the fractures.

A common problem is for a vertical borehole to intersect relatively few sub-vertical fractures. The Terzaghi correction in Equation 4.4 can be applied to calculate an appropriate adjusted number of fractures that accounts for the relative orientation between the rock face and the strike of the fracture:

$$N = \frac{N_a}{\sin\theta} \tag{4.4}$$

where N_a is the apparent number of fractures measured along the scan line and N is essentially a weighted number of fractures that accounts for the biased sampling.

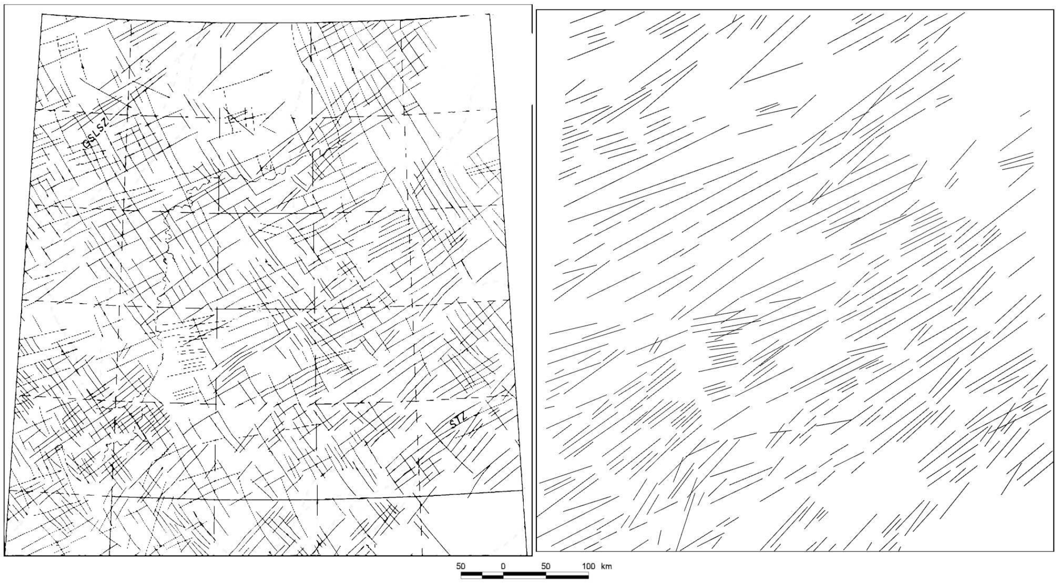


Figure 4.2: On the left: A map of lineaments from Northern Alberta (Pana et al., 2001). On the right: The southwest-northeast set is digitized.

4.3 Modelling Natural Fracture Networks with DFNs

4.3.1 A Map of Lineaments from Northern Alberta

Figure 4.2 shows a map of two lineament sets occurring in an area of Northern Alberta. A lineament is a linear topographic feature, which is an expression of an underlying geological structure such as a fault. Fracture and shear zones can also give rise to lineaments. The lineaments on Figure 4.2 were inferred from satellite imagery and digital elevation models (Pana et al., 2001).

In total, there are 425 lineaments from the SW-NE set that were digitized and shown on the right side of Figure 4.2. It is common practice to group fractures into sets of similar orientation and simulate separate DFNs for each fracture set (Clemo, 1994; Cooper, 2000; Narr et al., 2006; Makel, 2007). A rose diagram of the lineament orientations suggests that they are from a single set (Figure 4.3), which suggests that a single DFN may adequately model the lineaments.

Without any other available information, the lineaments are assumed to be sub-vertical and represent underlying faults or fractures. This seems reasonable because most NFR studies are concerned with sub-vertical fractures since horizontal fractures

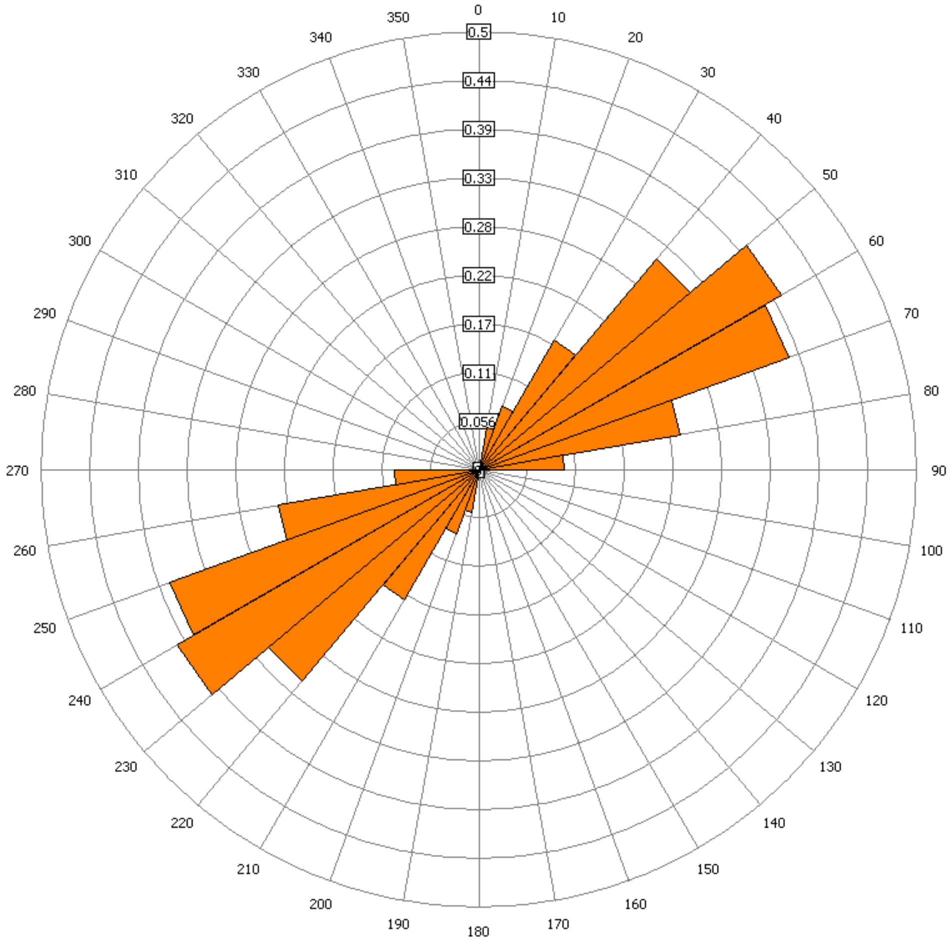


Figure 4.3: Rose diagram for the fractures shown in Figure 4.2.

would normally be closed due to the weight of the overburden. The digitized lineaments can be used to represent “the truth” and DFNs can be simulated as models of those lineaments. Then the spatial statistics of the digitized lineaments and the simulated DFNs can be compared.

4.3.2 Calculating the Spacing of a Natural Fracture Network

The spacing of the digitized lineaments can be calculated along imaginary scan lines projected through the network of lineaments. The scan lines are drawn perpendicular to the average orientation of the lineaments. The distance along the scan lines between intersections with lineaments are individual measurements of spacing (Figure 4.4). The Terzaghi correction can be applied as in Equation 4.3. One hundred scan lines seeded

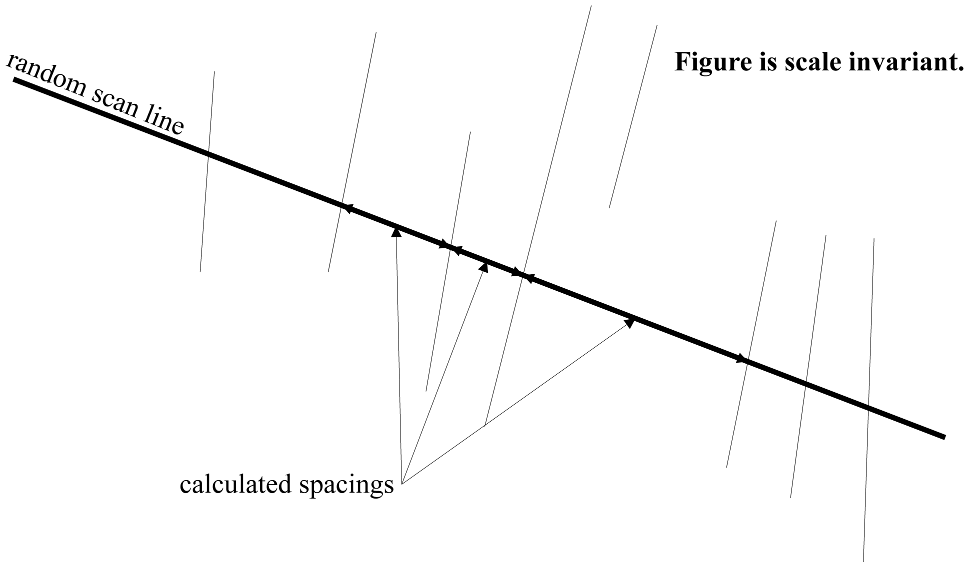


Figure 4.4: Measuring fracture spacing along a scan line that is perpendicular to the average fracture orientation.

from random locations within the map area are used to build a relative histogram of lineament spacing for the digitized map. The relative histogram is shown in Figure 4.5. The histogram shows that most lineaments have spacings between 3 and 30 kms.

4.3.3 Calculating Deviation in Local Orientation of Fractures in a Natural Fracture Network

The orientation of fractures in a network governs the direction of increased or decreased flow (depending on whether the fractures act as conduits or barriers to flow, respectively). An assessment of how similarly lineaments are oriented is required. Each lineament is visited and its nearest neighbour is identified *in the direction that is perpendicular to the lineament* (parallel to the lineament normal vector). The ***deviation in local lineament orientation*** is defined as the angle between the normal vectors of a lineament and its nearest neighbour:

$$\theta = \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right) \tag{4.5}$$

a and **b** are the normal vectors of two fractures. A relative histogram of the deviation in local lineament orientation is shown, for the digitized fractures, in Figure 4.6. the histogram shows that most lineaments are oriented within 12 degrees from their nearest

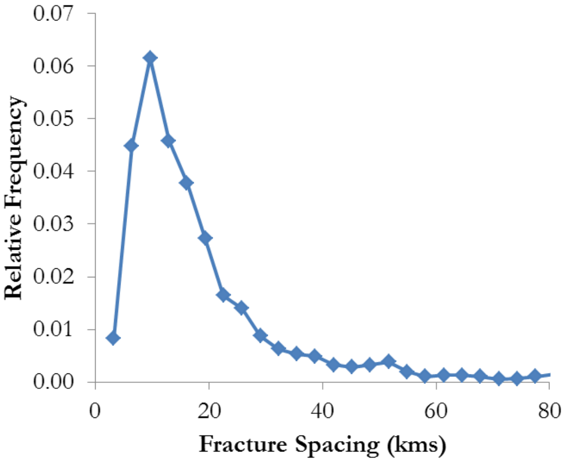


Figure 4.5: Relative histogram of lineament spacing for the digitized map (Figure 4.2).

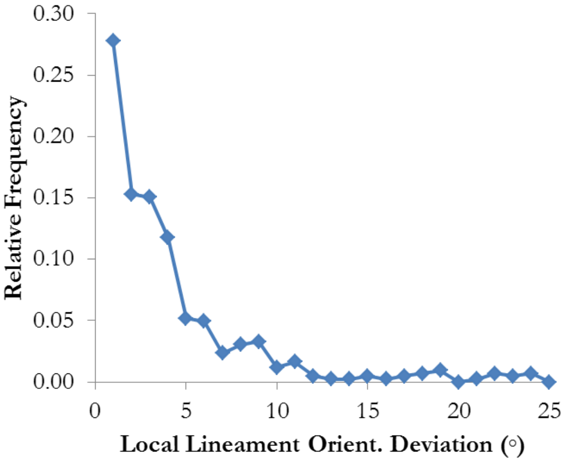


Figure 4.6: Relative histogram of deviation in orientation between nearby lineaments in the digitized map (Figure 4.2).

neighbour.

4.3.4 Modelling the Natural Fracture Network with DFNs

The next step is to simulate multiple realizations of DFNs as models for the digitized lineaments and assess how well their spatial statistics match. The DFNs were modelled with the Poisson process based approach, discussed at the start of the chapter, where fracture centroids are generated with a Poisson process and orientations are drawn independently of location. For simplicity, in this thesis, DFNs generated in this manner are referred to as **random DFNs**.

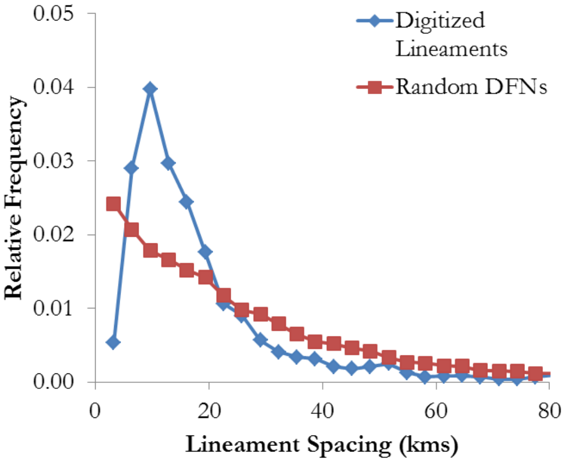


Figure 4.7: Relative histograms of spacing for the digitized lineaments Figure 4.2() compared to 100 realizations of random DFNs.

The algorithm for generating the random DFN realizations involves re-sampling the lineament orientations and centroid locations of the digitized lineaments:

1. visit a lineament randomly
2. re-sample its centroid x and y location from within the area of interest, defined by the limits of the lineament map
3. re-sample the lineament orientation (strike) from the PDF of lineament strike
4. go to 1, visiting a new lineament each time until every lineament has been visited
5. once all lineaments have been visited, the resulting DFN is one realization

Figure 4.7 shows a histogram of lineament spacing for 30 DFN realizations (red line with squares), compared to the histogram for the digitized lineaments (blue line with diamonds). The digitized lineaments have an average spacing of 17.7 km with a standard deviation of 17.4 km. The random DFN fractures have a larger average spacing of 24.1 km and a larger standard deviation of 23.4 km. Moreover, the spacing histogram for the digitized lineaments is narrower and less skewed than for the 30 realizations. The spacing bin with the highest frequency for the digitized lineaments is 13 km. By comparison, the most frequent spacing bin for the 30 DFN realizations

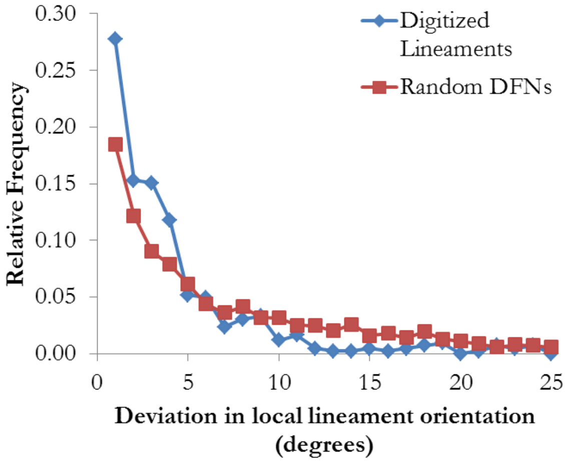


Figure 4.8: Relative histograms of the deviation in orientation between nearest neighbours, compared to 100 realizations of random DFNs.

is the 3 km bin. This result makes sense when you consider that the distribution of intervals between Poisson distributed events (centroids) is exponential (Belfield, 1998; Mitrani, 1982). Thus, Poisson distributed fractures should not normally be expected to reproduce a non-exponential distribution of fracture spacing.

With 425 lineaments per realization and 1000 random sampling lines per realization, the histogram of spacing converges quickly with increasing realizations. Using more than 30 realizations does not appreciably change the results.

Figure 4.8 shows the histograms of deviation in local orientation between lineaments for the random DFNs (red line with squares) and the digitized lineaments (blue line with diamonds). The histogram shows that almost all of the digitized lineaments are oriented within 12 degrees of their nearest neighbour. However, for the random DFNs, the histogram is much wider. For the random DFNs, most fractures are oriented within approximately 22 degrees of their nearest neighbour. The random DFNs show less of a tendency for fractures to be oriented similarly to their nearest neighbours than is shown by the digitized lineaments. Normally, a wider distribution of fracture orientations will lead to greater fracture connectivity created by more intersections between fractures.

Fracture network permeability is tied, in part, to the amount of connectivity between fractures. That is, all else being equal, well-connected fracture networks with many intersections are more permeable than discontinuous fracture networks with few

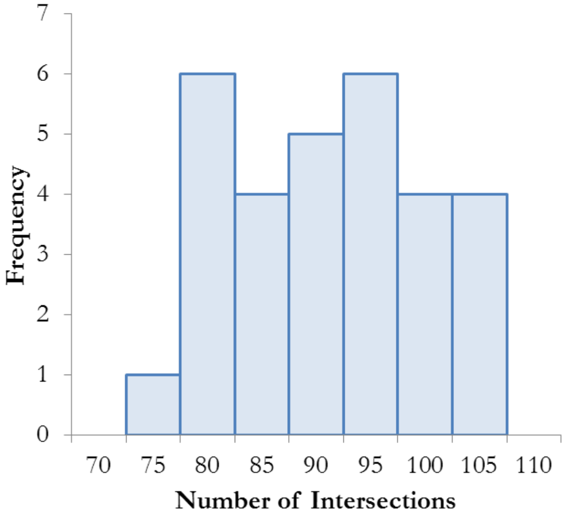


Figure 4.9: Histogram of number of intersections per realization.

intersections. Thus, it is important for a DFN to have a similar connectedness (defined here by the number of fracture intersections) compared to the natural fracture network it is designed to represent.

There are 20 intersections in the map of digitized lineaments. The number of fracture intersections was also calculated for each of the 30 random DFN realizations. The average number of fracture intersections per realization is 89.4 with a standard deviation of 8.4 intersections. The number of fracture intersections in the digitized fracture network is 7.5 standard deviations less than the mean for the 30 DFN realizations. The DFN with the fewest number of intersections still has 75 intersections. Figure 4.9 shows a histogram of the number of intersections per realization.

Based on this analysis, the spatial statistics of the digitized lineaments shown in Figure 4.2 cannot be realistically modelled with the typical random DFN. The DFNs do not honour the histograms of lineament spacing and deviation in local orientation. Moreover, the random DFNs showed far more intersections than were measured on the digitized map.

4.3.5 Rock Outcrop from Vernazza, Italy

Figure 4.10 shows a picture of an exposed rock face along a walking trail near Vernazza, Italy. There are two fracture sets in the rock face. The main fracture set shows three

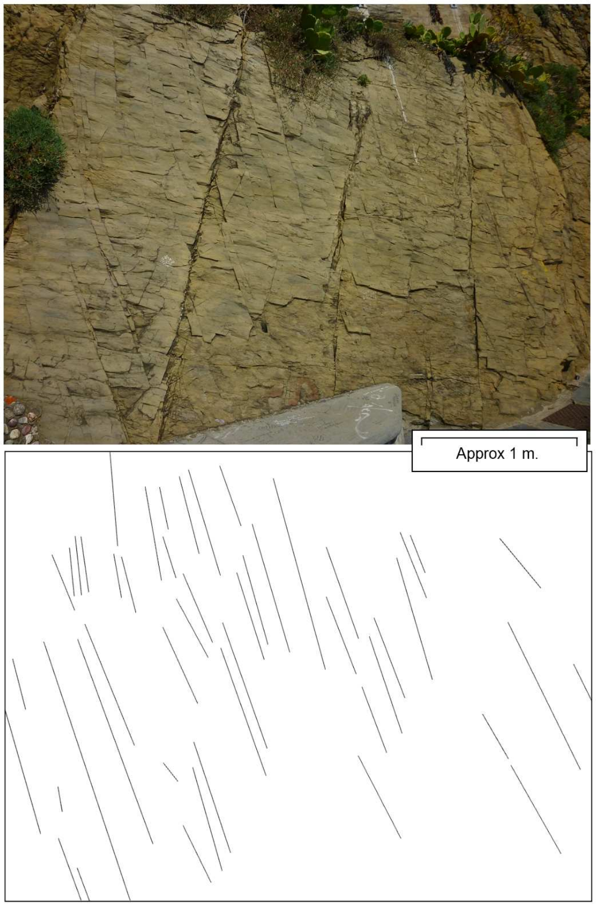


Figure 4.10: Above: An exposed rock face in Vernazza, Italy showing two fracture sets. Below: One of the fracture sets is digitized.

very large fractures that are very noticeable and dip towards the lower left corner of the photo. The secondary fracture set has many more fractures that are less well defined and dip towards the lower right corner of the photograph. The traces from the second fracture set were digitized and are also shown in Figure 4.10. In total, 48 fractures were digitized.

The actual orientations of the fractures were not measured and recorded. Thus, this example is assumed to be two-dimensional and the vertical direction (up) corresponds to North. The average fracture strike is 162 degrees (measured clockwise from the upwards on the image of the digitized fractures), with a standard deviation of 7.1 degrees. Based on visual inspection, the histogram of fracture orientation is approximately normal as is shown in Figure 4.11.

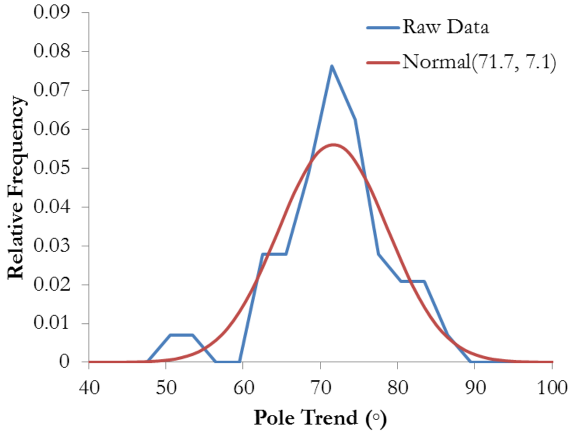


Figure 4.11: The fracture orientation (pole trend) data is approximately normally distributed.

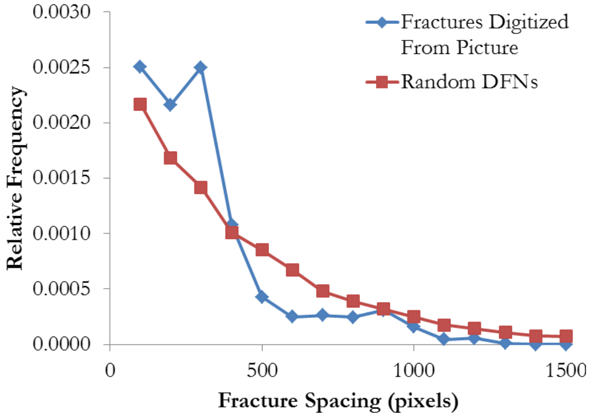


Figure 4.12: Histograms of fracture spacing for the digitized fractures (Figure 4.10) and the 100 random DFNs. The spacing is measured in units of pixels as there is no accurate scale for the photograph.

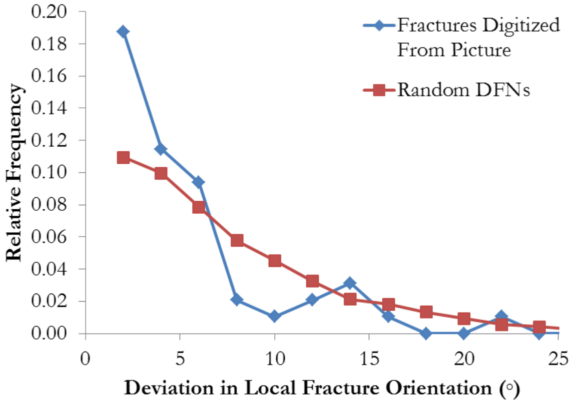


Figure 4.13: Histograms of deviation in local orientation for the digitized fractures (Figure 4.10) and the 100 random DFNs.

Figure 4.12 shows a relative histogram of fracture spacing for the digitized fractures (blue line with diamonds). As there is no accurate distance scale for the photograph, distance is measured in pixels. The figure also shows the relative histogram for 100 realizations of random DFNs created by the same re-sampling algorithm detailed in Section 4.3.4. The histogram for the digitized fractures shows a greater frequency of shorter fracture spacings (i.e. below 400 pixels). The random DFNs have a wider histogram of fracture spacing.

Figure 4.13 shows relative histograms of deviation in local orientation for the digitized fractures (blue line with diamonds) and for the 100 realizations of random DFNs (red line with squares). The histogram for the digitized fractures shows a higher incidence of low angles between nearby fractures compared with the histogram for the random DFNs. This means that the digitized fractures are oriented more similarly to their nearest neighbours than they would be if their orientation and location were independent.

The digitized fractures show zero fracture intersections. However, in the 100 random DFN realizations, fracture intersections are quite common. A histogram of the number of fracture intersections is shown in Figure 4.14. In 100 realizations of the random DFNs, there is an average of 3.9 intersections per realization with a standard deviation of 2.1 intersections per realizations. Out of 100 realizations, only two had no fracture intersections and one realization had as many as 14 intersections.

Taken together, the fracture spacing, deviation in local fracture orientation and fracture intersection data indicate that the fractures digitized from the rock face in Italy could not be modelled by DFNs that draw centroid locations with a Poisson process and independently of each other and their orientations.

4.4 A New Approach to DFN Simulation

Some natural fracture networks cannot be modelled with random DFNs while still respecting certain spatial statistics (noted earlier). This section introduces a new methodology for simulating DFNs and reviews alternative methodologies that were considered and rejected.

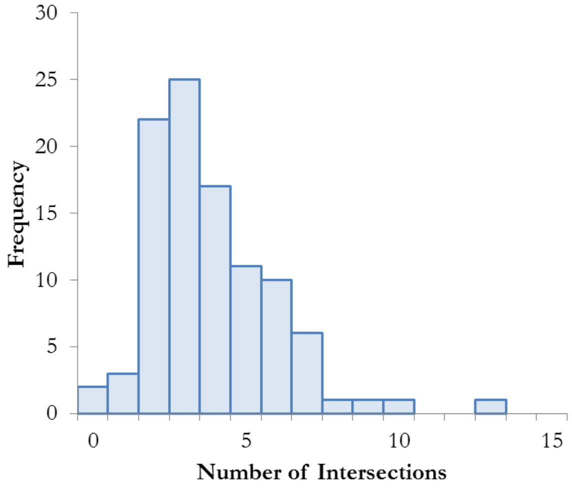


Figure 4.14: Histogram of the number of fracture intersections per realization for 100 realizations of random DFNs

4.4.1 Overview of Proposed Methodology

The proposed methodology works by simulating more fractures than are required and iterating to find a subset that closely match target spatial statistics.

The pool of all fractures is divided into two subsets; those that are part of the DFN and those that are not part of the DFN. Different subsets of fractures are tested by moving fractures into and out of the DFN (one at a time) and evaluating the effect of the change on the spatial statistics of the DFN.

The target spatial statistics are histograms of deviation in local orientation, fracture length and a measure of fracture spacing. Other target spatial statistics are a target number of fracture intersections and fracture intensity. An objective function, based on the squared difference between the target and actual fracture network spatial statistics, measures the impact of changes to the DFN. A greedy optimization algorithm was implemented to accept all positive changes to the DFN and reject all others. The choice of optimization method is explained in detail in Section 5.3.1 while rejected alternatives are discussed in Section 5.3.2.

The new methodology results in DFNs that are more geologically realistic since target spatial statistics are honoured.

The algorithm for the proposed algorithm is presented below:

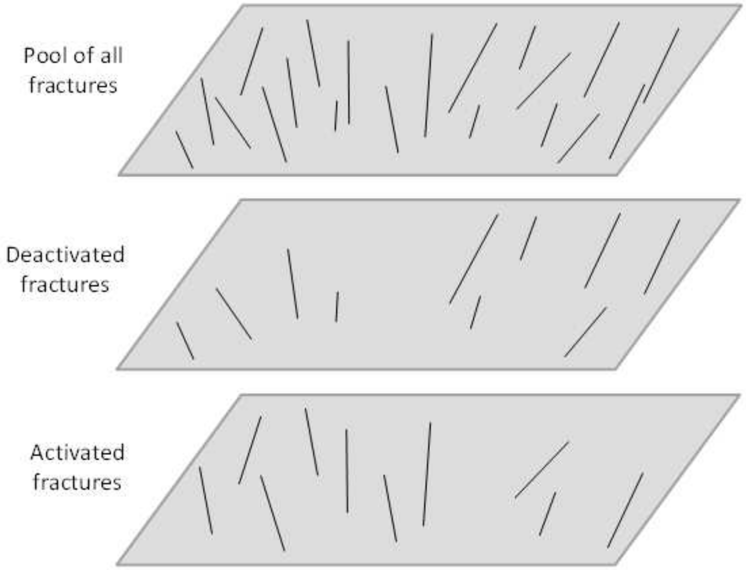


Figure 4.15: A 2D illustration of the pool of fractures, which consists of activated and deactivated fractures. There are 20 fractures in the pool. 10 fractures are activated and are the DFN at any time during the optimization process. 10 fractures remain deactivated and are not considered part of the DFN.

1. A pool of fractures is simulated using the traditional Poisson-based methods (i.e. random DFNs as discussed in Section 4.3.4). A central idea is to generate more fractures than are required. The ratio of the pool intensity to the target intensity is called the fracture multiplication factor (FMF).

2. Not all fractures are assigned to the DFN (see Figure 4.15). Some fractures are assigned to the DFN and are termed *activated* while fractures that are not part of the DFN are termed *deactivated*. This terminology will be explained as the rest of the methodology is presented.

3. An initial DFN is created by randomly visiting fractures in the pool and activating them to be part of the DFN. This process stops when the desired fracture intensity is achieved. Thus, the initial DFN closely matches the target fracture intensity before optimization begins.

4. A search strategy is implemented to discover the location of all fracture centroids with respect to all other fractures. The goal is to identify the fractures that are close to each other and calculate the distance between them as a measure of

fracture spacing. The angles between nearby fractures are also calculated; this is called the deviation in local fracture orientation as was described earlier.

5. Intersections between the fractures in the pool are calculated and saved in memory. If an intersection exists between two fractures that are activated, that is also noted in memory.
6. Relative histograms of local fracture spacing, deviation in local fracture orientation and fracture length are calculated.
7. An objective function is calculated for the initial DFN. The objective function measures the difference between actual and target histograms of local fracture spacing, deviation in local fracture orientation, fracture length, fracture intensity and the actual and target number of fracture intersections. This is discussed in more detail in Section 5.3.
8. A random path to visit each fracture in the pool is determined.
9. The initial DFN is iterated upon by visiting a fracture on the random path and switching its activation. If the fracture is activated (currently part of the DFN), then it is deactivated and removed from the DFN. If the fracture is deactivated (not currently part of the DFN), it is activated and inserted into the DFN.
10. The objective function is re-calculated for the modified DFN.
11. A greedy optimization algorithm (see Section 5.3.1 for details) was used where the change to the DFN is accepted if the objective function decreases and rejected otherwise.
12. The process repeats, visiting a new fracture each time (i.e. return to step 9) until all fractures in the pool are visited once.
13. Additional improvement in the optimization is made by looping over each fracture a number of times (i.e. return to step 8), with a new random path each time.

The methodology is not entirely statistical and the result is a meta-heuristic algorithm to generate DFNs that are not entirely random in nature, by reproducing information that is calibrated from available measurements and analogue sites that are deemed representative. There are no geomechanical principles directly employed in the generation of the fracture networks. Note that the algorithm is general enough to work in two or three dimensions.

4.4.2 Alternatives to the Proposed Methodology

This chapter has demonstrated the need for a new algorithm capable of generating more geologically realistic DFNs. The algorithm proposed in the previous subsection is one of many possibilities. In fact, several other approaches to DFN simulation were considered. This subsection briefly discusses those alternative approaches and whey they were not pursued further.

Fracture Network Perturbation

An alternative methodology would be to simulate the correct fracture intensity (i.e. no extra fractures) and perturb the fracture network by visiting a fracture at random and re-simulating its centroid location, fracture orientation and size. Then the change in the objective function could be observed. If the objective function decreases as a result of the change, the change would be accepted.

This approach would produce similar results compared to the methodology proposed in this thesis. Perturbing a fracture is comparable to removing a fracture from the DFN and inserting a new fracture into the DFN.

One major disadvantage of the perturbation approach is that it will produce DFNs that *must* match the target fracture intensity. Alternatively, the algorithm proposed in this thesis allows the fracture intensity to depart from the target if it results in an improvement to the DFN in other ways (i.e. if the changes result in better matches for fracture spacing or intersections).

The perturbation approach is also less computationally efficient than the methodology proposed in this thesis. Consider the following example. Under the proposed approach, if the target intensity is 250 fractures and the FMF is two, then there are 500 fractures in the pool. If the number of iteration loops is two, then 1000 changes to the DFN are considered with only 500 data searches. If the perturbation approach is considered, only 250 fractures are generated. To achieve 1000 changes to the DFN, 1000 data searches are required.

Direct Statistical Approach

It would be impossible to directly simulate the locations and attributes of potentially millions of fractures because there is no evident way of constructing a multivariate distribution that could be sampled in a traditional way. One would somehow need to define a multivariate distribution of fracture locations, relative orientations, length, intersections and size in order to obtain a fracture network that honoured those distributions.

One might be able to imagine a method for sequentially simulating fracture locations given previously simulated fractures. If a probability distribution of fracture spacing can be defined, it might be possible to draw fracture spacings rather than fracture locations. Then, the newly simulated fracture could be located at that spacing from other fractures in the vicinity. For example, if it is assumed that one fracture has already been simulated, then one could randomly draw from a distribution of spacing and locate the new fracture at a position corresponding to the drawn spacing relative to the first fracture. However, it is not clear what side of the first fracture the second fracture should be placed. Also, it is not clear where the second fracture should be located in three-dimensional space. The spacing only addresses the distance between the fractures in the perpendicular direction. The centroid of the second fracture is "free to move" in the plane that is parallel to the first fracture. Things get even more complicated when a third fracture is simulated. If it is placed in between the first and the second fractures, then the originally drawn spacing is compromised.

Direct Geomechanical Approach

Direct geomechanical techniques use the estimated current or inferred past stress state in the reservoir rock mass to determine the orientation and density of fractures with respect to a defined fault framework. Displacement or stress state boundary conditions are applied to the model, resulting in a strain tensor which indicates fracture orientation and density. Such geomechanical approaches have been tried in the past (Bourne et al., 2000; Renshaw and Pollard, 1994; Heffer et al., 1999) and are computationally expensive, even in two dimensions. Fracture network simulation by geomechanical principles has not been successfully extended to three dimensions, especially for reservoir scale models (Srivastava, 2006). An additional complication is that the stress state and rock strength parameters are very imporant to the results yet are poorly constrained (Makel, 2007).

Pseudo-Geomechanical Approach

One could consider some type of pseudo-geomechanical approach, similar to that presented by Srivastava (2006). A pseudo-geomechanical approach might attempt to mimic the direct geomechanical approach of Renshaw and Pollard (1994) without directly applying geomechanical modelling. Srivastava's approach replaces geomechanical principles for fracture propagation with geostatistical rules, which are able to produce some of the geological realism with less computational effort. The main difficulty would be in defining the rules for propagation of the fractures. Srivastava's work requires detailed joint mappings from outcrops, whereas the proposed methodology does not require outcrops. Srivastava's work also randomly seeds fracture locations.

Sequential Rejection Sampling

It might be possible to simulate fracture networks by rejecting fractures sequentially as they are simulated. For example, a newly simulated fracture might be rejected if a new fracture:

- is simulated too close to an existing one;
- has an orientation that is too dissimilar to its nearest neighbour;

- intersects an existing one at a very low angle.

If a probability distribution function (PDF) for fracture spacing can be defined, then a simulated annealing-like approach might be considered where a new fracture with a particular spacing to an existing fracture could be rejected with a certain probability depending on its location under the PDF.

The main challenge with this methodology is that it would be difficult to control the resulting fracture spacing and orientation distributions. For example, imagine two fractures, A and B, that have already been simulated. Now imagine that a third fracture, C, is inserted into the DFN in between fracture A and B. Fracture C might be accepted based on its spacing to fracture A, but now the spacing between A and B is irrelevant even though B was accepted based on its distance to fracture A.

Similarly, it would also be near impossible to control the number of intersections. Early in the simulation, the DFN has less than the target amount of intersections and no newly added fractures would be rejected based on its intersections with existing fractures. However, when the number of intersections equals the target number, all new fractures intersecting existing fractures would be rejected. One could consider employing a probabilistic approach to rejection where new fractures that intersect others are rarely rejected in the early stages. Then as the number of intersections approaches the target number, rejections become more common. However, this might leave artefacts in the fracture networks since most intersecting fractures would be developed early in the DFN simulation process.

4.5 Remarks

The purpose of this chapter was to introduce the typical DFN modelling algorithm (i.e. "the random DFN") and assess how well it works at generating models for existing natural fracture networks. Fracture intensity, local fracture spacing, deviation in local fracture orientation and the number of fracture intersections are introduced as key fracture network spatial statistics that should be honoured in DFNs. Two real-life examples of natural fracture networks were presented and it was shown that the DFNs generated by the typical random DFN algorithm do not honour the spatial statistics of the natural fracture networks they attempt to model. A new method for simulating DFNs was proposed that is designed to honour certain target spatial statistics. The DFN simulation algorithm was codified in a `FORTRAN` program called `DFNSIM` for **d**iscrete **f**racture **n**etwork **sim**ulation, which is discussed in greater detail in the next chapter. While this chapter introduced the proposed DFN simulation algorithm, the next chapter discusses several key implementation considerations.

Chapter 5

Implementing the Proposed DFN Simulation Algorithm

Chapter 4 showed that DFNs generated using traditional methods (i.e. the "random DFN") may be unsuitable for developing models of some natural fracture networks. A new algorithm was proposed for simulating DFNs that match input target spatial statistics.

This chapter discusses several key aspects of implementing the algorithm including: 1) how the pool of fractures is generated, 2) calculating the spatial statistics of a simulated DFN, 3) the objective function, 4) the data search strategy, 5) issues of convergence 6) measures of fracture spacing and 7) other parameters and the `DFNSIM` parameter file.

5.1 Generating a Pool of Fractures and an Initial DFN

Generating a large pool of fractures, that exceeds the target fracture intensity, permits an optimization to find a suitable subset that comes close to matching target fracture network spatial statistics. The choice of pool size is subjective but testing has shown that a pool intensity that is two to three times the target intensity is usually sufficient. The pool of fractures can be generated using the traditional random DFN approach:

1. A fracture centroid location (x, y, z) is generated with a Poisson process. For DFNs where the target intensity varies little over the modelling area, a homogeneous underlying intensity function is appropriate. If the target fracture intensity

varies greatly over the modelling area, a non-homogeneous underlying intensity function may be used.

2. A fracture orientation is drawn independently of the centroid location. The orientation is drawn from an input distribution derived from available data (orientation measurements from core, FMI, outcrops, areal photos, etc.).

3. Fracture length, height and aperture are drawn independently from separate input distributions. These distributions are also derived from data (where possible) or analogues.

4. Every fracture in the pool (Figure 4.15) starts in the deactivated state (i.e. not part of the DFN). An initial DFN is created by randomly visiting fractures in the pool and activating them, which assigns them to the DFN. The initial DFN is complete when its intensity closely matches the target fracture intensity. If a P_{30} intensity (fracture count per volume) is used, the initial DFN will have the exact right intensity, since it is just a matter of activating the correct number of fractures. If the P_{32} intensity (fracture area per volume) is used, fractures are activated until the DFN intensity just exceeds the target intensity.

This approach results in an initial DFN that would be similar to those created by commercial software using the Baecher model.

5.2 Calculating the Spatial Statistics of the DFN

After the pool of fractures is generated and a random subset has been assigned to the initial DFN, an objective function is calculated to measure the “goodness of fit” between the target input histograms and statistics and those of the initial DFN. The objective function is based, in part, on histograms of local fracture spacing and deviation in local fracture orientation.

A geologist or engineer might measure fracture spacing using scan lines (Figure 4.4). However, when generating and optimizing large fracture networks this approach was found to be too computationally expensive. One could imagine a fracture net-

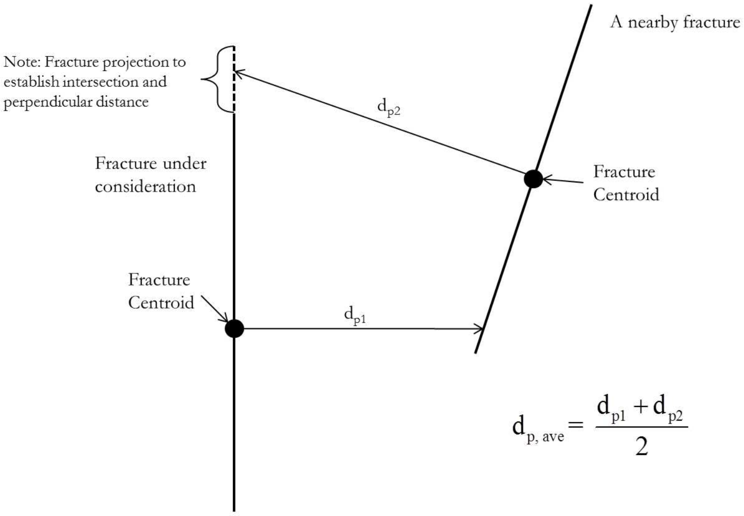


Figure 5.1: Calculating the perpendicular distance to the nearest fracture.

work with millions of fractures and thousands (or more) imaginary scan lines through which fracture spacing is calculated. Each time a fracture is removed or added over potentially millions of iterations, the true spacing would need to be re-calculated or updated somehow. Early attempts at implementing such a procedure proved too computationally expensive, limiting the practical size of DFNs to the order of thousands of fractures. Thus, an alternative measure of fracture spacing was required.

The average perpendicular distance between a fracture and its nearest neighbour is proposed as a measure of local fracture spacing. This agrees with the definition of fracture spacing by Makel (2007) as the orthogonal distance between fracture planes. Figure 5.1 shows the calculation of the perpendicular distance between a fracture and its nearest neighbour. Each fracture is visited and its nearest neighbour is identified. Fracture spacing is usually measured normal to the plane of the fracture. Thus, the perpendicular distance between fractures is used. The perpendicular distances (d_{p1} and d_{p2}) between each fracture's centroid and the other fracture are calculated. The average of d_{p1} and d_{p2} is taken to be the local fracture spacing since d_{p1} and d_{p2} are usually not equal. If the ray from one centroid does not intersect the other fracture, the intersection is taken where it would have been if the fracture was infinite in extents (see Figure 5.1).

In addition to local fracture spacing, the similarity in orientation between nearby fractures is computed. The deviation in local fracture orientation is calculated by finding the angle between the poles of two nearby fractures and is calculated from the dot product of the two normal vectors as in Equation 4.5.

All fractures are visited and local fracture spacing and deviation in local fracture orientation are calculated. Histograms of both variables are constructed.

5.3 The Objective Function

Now the objective function can be calculated. The objective function measures the squared difference between the bins of target histograms and those calculated from the DFN. The objective function is:

$$\begin{aligned} O &= \sum_{c=1}^{5} C_c O_c \\ &= C_{spac} \sum_{i=1}^{sbins} \left(S_i^{target} - S_i^{DFN}\right)^2 + C_{or} \sum_{j=1}^{orbins} \left(Or_j^{target} - Or_j^{DFN}\right)^2 \\ &\quad + C_{len} \sum_{k=1}^{lbins} \left(L_k^{target} - L_k^{DFN}\right)^2 + C_{inter} \left(Inter^{target} - Inter^{DFN}\right)^2 \\ &\quad + C_{int} \sum_{l=1}^{ibins} \left(I_l^{target} - I_l^{DFN}\right)^2 \end{aligned} \tag{5.1}$$

where:

- S_i^{target} and S_i^{DFN} are the target and DFN histograms of local fracture spacing, respectively. $sbins$ is the number of histogram bins.

- Or_i^{target} and Or_i^{DFN} are the target and DFN histograms of deviation in local fracture orientation, respectively. $orbins$ is the number of histogram bins.

- L_i^{target} and L_i^{DFN} are the target and DFN histograms of fracture length, respectively. $lbins$ is the number of histogram bins.

- $Inter_i^{target}$ and $Inter_i^{DFN}$ are the target and DFN number of fracture intersections, respectively.

- I_i^{target} and I_i^{DFN} are the target and DFN histograms of fracture intensity, respectively. $ibins$ is the number of histogram bins.
- C_{spac}, C_{or}, C_{len}, C_{inter}, and C_{int} are coefficients, which allow each component to play an equally important role in the objective function and serve to make it unit-less in order to compare components with different original units.

The coefficients are calculated automatically so that, on average, each component contributes equally to changes in the objective function. This can be achieved if each of the coefficients are inversely proportional to the average change of that component objective function (Deutsch, 1992).

The change in the objective function due to a fracture activation or deactivation is:

$$\Delta O = O_{new} - O_{old} = \sum_{c=1}^{5} C_c \Delta O_c \tag{5.2}$$

where c represents the five different components of the objective function. Then the coefficients are:

$$C_c = \frac{F_{c,Scaling}}{\overline{|\Delta O_c|}}, \quad c = 1, ..., 5 \tag{5.3}$$

where $F_{c,Scaling}$ are scaling constants. The scaling constants allow the user to emphasize or de-emphasize the importance of certain components of the objective function as necessary.

The average change in each component of the objective function, ΔO_c, can be approximated by evaluating the average change of N changes (say $N = 1000$) to the DFN. The procedure is: 1) visit a fracture randomly, 2) change its activation state, 3) observe the effect of the change on the objective function, 4) reverse the change to the DFN, and 5) repeat, visiting a new fracture each time until N changes are made. The average change for each component is:

$$\overline{|\Delta O_c|} = \frac{1}{N} \sum_{i=1}^{N} \left[O_c^{base} - O_c^{i} \right], \quad c = 1, ..., 5 \tag{5.4}$$

where $\overline{|\Delta O_c|}$ is the average change for component c, O_c^{base} is the objective function for the initial DFN (i.e. the base case) and O_c^i is the objective function for the perturbed DFN. The base objective function, O_c^{base}, is calculated with the five coefficients equalling 1.

Once the coefficients, C_c, are determined, the proposed fracture simulation algorithm proceeds to steps 6 and 7 (Section 4.4), where the initial DFN is iterated upon and changes to the DFN are accepted or rejected depending on whether or not the change results in a decrease in the objective function. The objective function is flexible and can optimize the fracture network on any or all of its five components. For example, information on fracture length is difficult to obtain in a petroleum reservoir setting where fractures are sampled by core. If there is great uncertainty in the fracture length histogram, the fracture modeller could choose not to optimize on fracture length.

5.3.1 The Greedy Optimization Process

A greedy optimization is implemented in `DFNSIM`. That is, changes to the DFN are always accepted if they result in a decrease in the objective function result.

The greedy optimization was chosen because it is simple both conceptually, practically (it was easy to encode into `DFNSIM`) and since it performed well. The reason the greedy algorithm works well is that convergence is generally easy, insofar as excellent results were obtained for examples shown in this thesis (specific convergence results are discussed in Section 5.6.2).

It is possible that the greedy algorithm might not be suitable for all cases. Two circumstances that might result in poor results with the greedy algorithm are: 1) If additional variables are added to the objective function, and 2) if the target spatial statistics are jointly unrealistic.

Future work may modify the proposed DFN simulation algorithm by adding additional components to the objective function such as the type and proportion of intersection truncations (i.e. fractures terminating at their intersection with another fracture). Adding more components to the objective function is likely to make the optimization more difficult.

Target spatial statistics (i.e. histograms of local fracture spacing, deviation in local fracture orientation and fracture length along with the number of fracture intersections and fracture intensity) must be defined prior to using `DFNSIM`. With limited data on the underlying natural fracture network, it is possible to define incompatible target spatial statistics. For example, it seems reasonable to say that there is a relationship between the fracture intensity, the fracture length distribution, the orientation distribution and the resulting number of fracture intersections. As fracture lengths increase, the number of fracture intersections will increase (assuming all other fracture size and orientation parameters remain constant). Similarly, as the fracture intensity increases, the number of fracture intersections will also increase (again, with all other size and orientation parameters remaining constant). Thus, one can imagine a situation in which the target fracture intensity, fracture length distribution and fracture orientation distribution could be defined in a way that makes it impossible to achieve a certain number of fracture intersections.

In other cases, it might be that the greedy algorithm may reach a local minima before an acceptable convergence is achieved. In cases where convergence to an acceptable solution is difficult, it may make sense to consider an alternate optimization algorithm.

5.3.2 Other Optimization Algorithms

One classic way of optimization would be to derive the gradient of the function to be optimized and employ a gradient descent or quasi-Newton method. However, in this case, the function is not known a priori making these methods intractable.

Meta-heuristic methods, such as the one proposed in this thesis, seem well suited for this type of combinatorial optimization problem as they are often efficient for solving large and complex problems (Talbi, 2009).

The simplest algorithm to consider is the brute force algorithm, whereby all possible combinations of subsets of fractures are evaluated and the subset with the lowest objective function result is the solution. The brute force algorithm will always produce the best result (the global minimum objective function), but is intractable for any practical scenario. If there are N fractures in the pool, then there are $2^N! - 1$ (the case where the DFN has zero fractures is excluded) possible combinations. Infeasible scenarios could be excluded outright. For example, if the target fracture intensity is N_{target} fractures, any scenarios outside the interval of $N_{target} \pm T_{tolerance}$ could be discarded, where $T_{Tolerance}$ is a tolerable deviation from the target fracture intensity. Even though this would greatly reduce the number of combinations to try, the problem is still intractable for all realistic reservoir scale fracture networks.

A simulated annealing (Kirkpatrick et al., 1983) optimization algorithm could be used instead of the greedy algorithm. If a change to the DFN results in an improvement to the objective function, the change would be accepted. Changes to the DFN that do not result in improvement to the objective function would also be accepted a certain percentage of the time. The simulated annealing approach avoids the problem of "getting stuck" in a local minimum.

Genetic algorithms (GA's) are another family of meta-heuristic algorithms that could have been considered for this optimization problem. GA's are search heuristics that mimic the process of evolution for the purposes of generating solutions to optimization and search problems. It may be possible to use a GA to optimize fracture networks in the context of the proposed algorithm, although this is more complicated than the greedy method implemented.

5.4 Data Search Strategy

Once the initial DFN is created, a search is required to locate all fracture centroids, calculate the distance between nearby fractures and identify which fractures are closest to each other. Specifically, the perpendicular distances between fractures are needed as a measure of fracture spacing. Once the perpendicular distances are known, its histogram can be calculated.

The simplest search strategy is to systematically calculate all perpendicular distances between all n fracture centroids. Unfortunately, the perpendicular distance calculation is much less efficient than a straight-line Euclidean distance. A more efficient approach is to use a search algorithm like the super block search (Deutsch and

Journel, 1998) to calculate distances between nearby centroids with a Euclidean distance first. Then some number (*nclose*) of fracture centroids that fall within a certain radius are subject to the perpendicular distance calculation (See Figure 5.2).

The data search procedure is as follows:

1. Set up and initialize the super block search.
2. Visit a fracture f_i $(i = 1, ..., n)$
3. The superblock search is used to identify the nearest *nclose* fractures to centroid f_i and how far they are away (by Euclidean distance). A radius is specified so that $nclose < n$.
4. The perpendicular distances between centroid f_i and the other *nclose* centroids are calculated and sorted. The fracture indices for the nearest *keepn* centroids are stored in memory a with their associated perpendicular distances.

`DFNSIM` is designed so that the search runs once after the creation of the initial DFN. Since all required nearby fracture indices and their perpendicular distances are stored, there is no need to recalculate any distances as fractures are activated and deactivated when the DFN optimization is under way.

5.4.1 Perpendicular Distance Calculation

The super block search calculates distances between centroids using a Euclidean distance. However, a perpendicular distance between fractures is more reflective of true fracture spacing. This section reviews the calculation of the perpendicular distance.

Calculating the perpendicular distance between two fractures is akin to calculating the shortest distance between a point (i.e. a fracture centroid) and an infinite plane (i.e. another fracture) (See Figure 5.3). Fractures are not infinite in extents, but for the calculation it is convenient to make that assumption.

Fracture A in Figure 5.3 is given by:

$$ax + by + cz + d = 0 \tag{5.5}$$

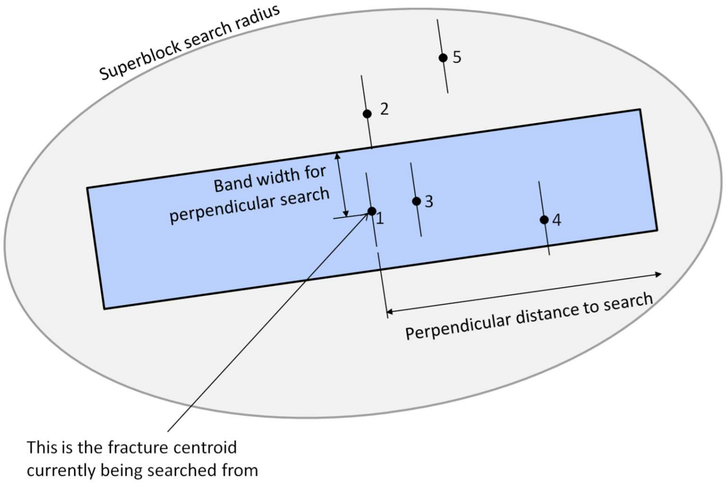


Figure 5.2: The relationship between the superblock search and the perpendicular search. The program is currently at location 1 and is searching for other fractures nearby. The super block search identifies all five fracture centroids within the search radius. However, the perpendicular distance is only calculated for the first four centroids due to the choice of band width and perpendicular distance to search. Note that fracture 3 is the closest to fracture 1 using Euclidean distance, but when perpendicular distance is used fracture 2 is closer to 1 than 3. A bandwidth is used to trim fractures that are far away in the direction of fracture strike.

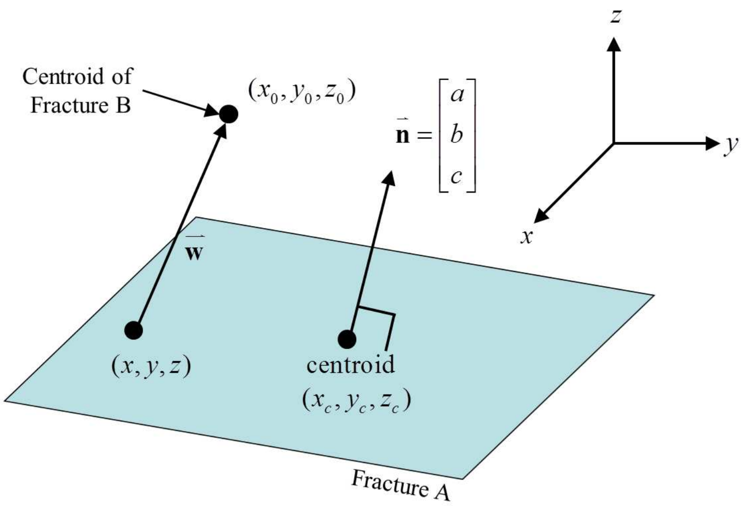


Figure 5.3: Illustration of the point to plane distance calculation. Figure is not to scale.

where a, b and c are the components of the normal vector to the plane, $\overrightarrow{\mathbf{n}}$. Since a, b and c are known along with the centroids coordinates, x_c, y_c and z_c, d can be calculated:

$$d = -ax_c - by_c - cz_c \tag{5.6}$$

The point (x_0, y_0, z_0), is the centroid of another fracture (Fracture B). Thus the perpendicular distance from fracture A to fracture B is the minimum distance between (x_0, y_0, z_0) and fracture A.

The vector, $\overrightarrow{\mathbf{w}}$, from the some unknown point on the plane, (x, y, z), to the fracture B centroid is given by:

$$\overrightarrow{\mathbf{w}} = -\begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} \tag{5.7}$$

If $\overrightarrow{\mathbf{w}}$ is projected onto $\overrightarrow{\mathbf{n}}$, we have the distance between fracture A and the centroids of B as:

$$D_{a-b} = \frac{|\overrightarrow{\mathbf{n}} \cdot \overrightarrow{\mathbf{w}}|}{|\overrightarrow{\mathbf{n}}|} \tag{5.8}$$

$$= \frac{|a(x - x_0) + b(y - y_0) + c(z - z_0)|}{\sqrt{a^2 + b^2 + c^2}} \tag{5.9}$$

$$= \frac{|ax + by + cz - ax_0 - by_0 - cz_0|}{\sqrt{a^2 + b^2 + c^2}} \tag{5.10}$$

By substituting in Equation 5.5, the perpendicular distance from between fracture A and B is:

$$D_{a-b} = \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{5.11}$$

The distance is calculated two ways: 1) from fracture A to the fracture B's centroid, and 2) from fracture B to the fracture A's centroid. Note that D_{a-b} is usually not equal to D_{b-a} (See Figure 5.4). The two distances are averaged to determine the average perpendicular distance between fracture A and B.

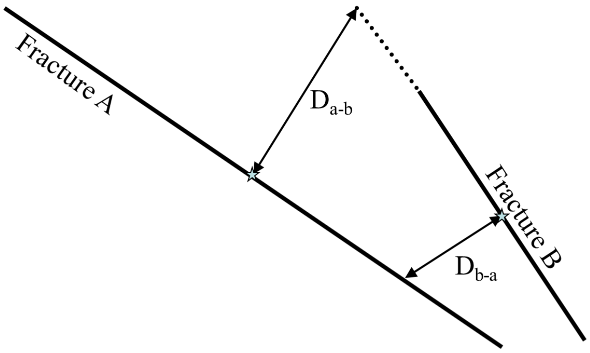


Figure 5.4: D_{a-b} is normally not equal to D_{b-a}. The fracture centroids are indicated by stars. Figure is not to scale.

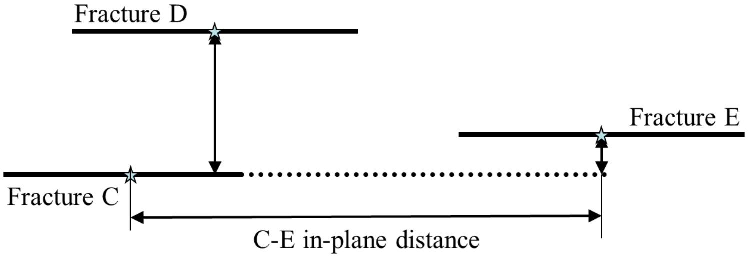


Figure 5.5: The perpendicular distance (indicated by arrows) between E and C is smaller than the perpendicular distance between D and C, even though D is much closer by Euclidean distance. Figure is not to scale.

$$D_{AB,ave} = \frac{D_{a-b} + D_{b-a}}{2} \tag{5.12}$$

As noted, the fractures are not infinite planes in reality. This leads to an important consideration as shown in Figure 5.5. The figure shows the perpendicular distance from D to C and from E to C. The perpendicular distance from E to C is less than that of C to D. It seems reasonable that the fracture spacing between D and C should be calculated. However, as the C-E in-plane distance increases, a fracture spacing between E and C makes less and less sense.

This becomes a problem if perpendicular distances are calculated for all fractures in the field. Fractures will be identified as very close (by small perpendicular distances) even if they are actually very far apart in the in-plane direction. The solution is to impose a bandwidth around fracture C (See Figure 5.6). and only consider fractures that occur within the bandwidth. In the example, the perpendicular distance is not

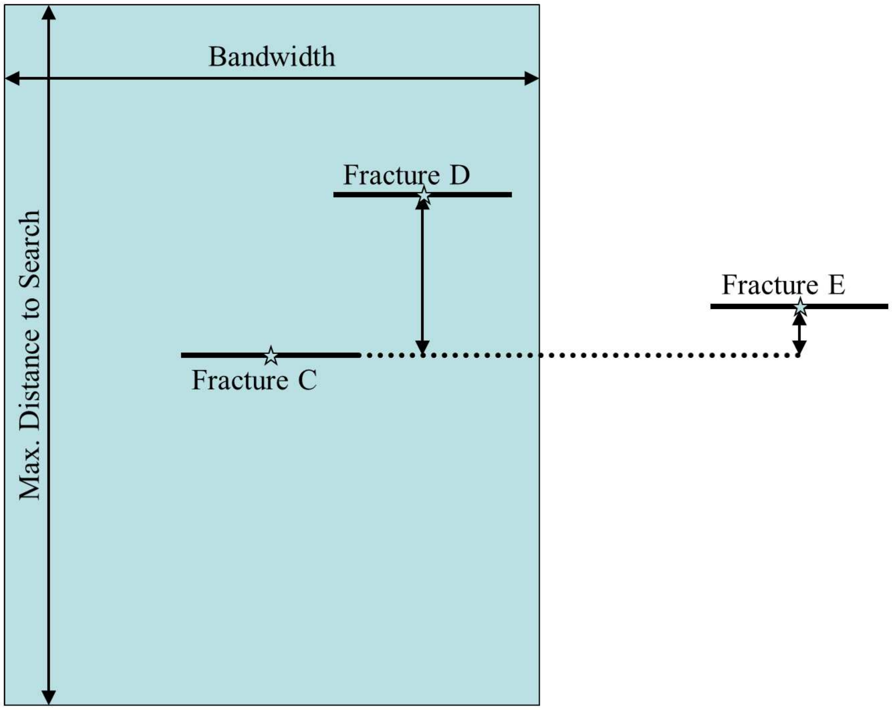


Figure 5.6: A bandwidth and maximum distance to search restricts fractures considered in the perpendicular distance calculation. Figure is not to scale.

calculated between fracture E and C because it is beyond the specified bandwidth from fracture C.

Similarly, a maximum distance to search is also imposed (See Figure 5.6). Beyond some distance, it is not relevant to consider the spacing between fractures. The `DFNSIM` user specifies a target histogram of local fracture spacing for which there is a maximum spacing bin. It is suggested that the maximum distance to search could be set to be just greater than the largest bin of the local fracture spacing histogram. After all, any fractures identified at further distances will not show up in the histogram.

One disadvantage to using the average perpendicular distance between fractures as a measure of fracture spacing is that the user-specified bandwidth and, to a lesser extent, the maximum perpendicular distance to search are parameters that must be determined. If the bandwidth is too large, `DFNSIM` may identify the nearest fracture as having a very short perpendicular distance even though it is far away in the direction of the fracture strike (Figure 5.5 and Figure 5.6). If the bandwidth is too small, some fractures may be ignored when they should not be. Additionally, if too few fractures fall within the bandwidth, they may not adequately describe a local fracture spacing histogram. It may be difficult to define what appropriate values for these parameters might be.

Figure 5.7 illustrates the problem. The figure shows a two-dimensional DFN with 10 fractures, which are numbered from 1 to 10. The fractures are shown in 2D as black lines with centroids locations indicated by filled circles. The red arrows point from one centroid to another, indicating which fracture is nearest to another fracture (measured by perpendicular distance). The fracture at the arrowhead represents the nearest fracture to the fracture at the arrow tail. For example, fracture 7 is the nearest to fracture 6 as is indicated by the arrow pointing from fracture 6 to fracture 7. To calculate the nearest fractures, a bandwidth larger than the field size was used. The problem is that very small perpendicular distances are calculated for centroids that are fairly far apart by euclidean distance. Note, that the arrows don't indicate the magnitude of perpendicular distance, only which fracture is the closest to another fractures. As is illustrated by the figure, if the bandwidth is too wide, the nearest fractures and their associated perpendicular distances do not make sense as a measure of fracture spacing. For example, fracture 3 and 4 are "nearest" to each other when a large bandwidth is used, even though the fractures are far apart along their strike direction. Measuring fracture spacing between fractures 3 and 4 makes little sense.

Figure 5.8 shows the same 10 fractures. However in this case, the nearest fractures are calculated with a bandwidth that is equal to the average fracture length (6 m). In this case, the nearest fractures make more sense when qualitatively compared with the notion of perpendicular distance as an alternative measure of fracture spacing. In this example, it seems to make more sense that fracture 5 is the closest to fracture 4, compared with fracture 3 in the previous example. Likewise, it also makes more sense that fracture 2 is the nearest fracture to fracture 3, rather than fracture 4 as in the previous example.

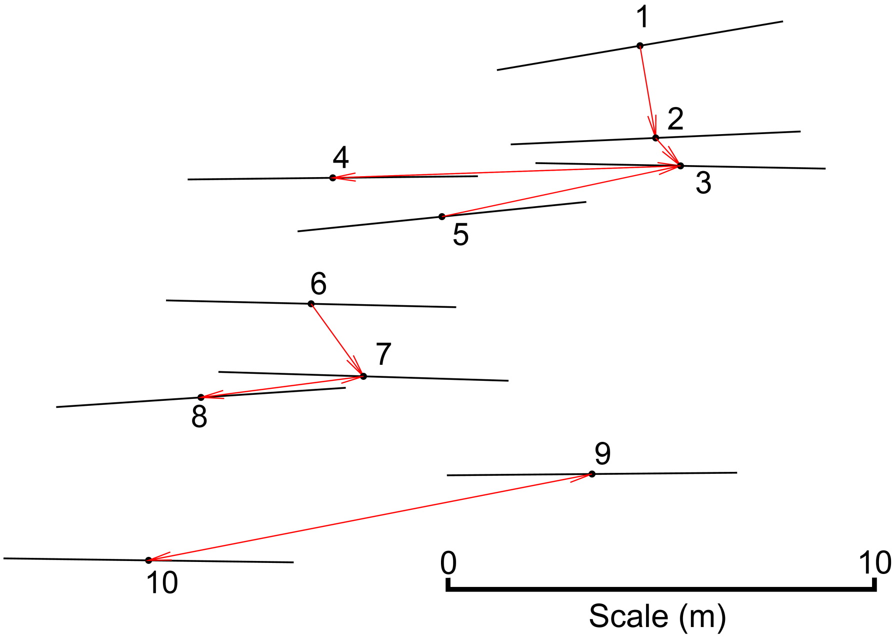


Figure 5.7: 10 two-dimensional fractures with nearest fractures indicated by the arrows. Bandwidth is set as larger than the field size. The red arrows point from a fracture to its nearest neighbour.

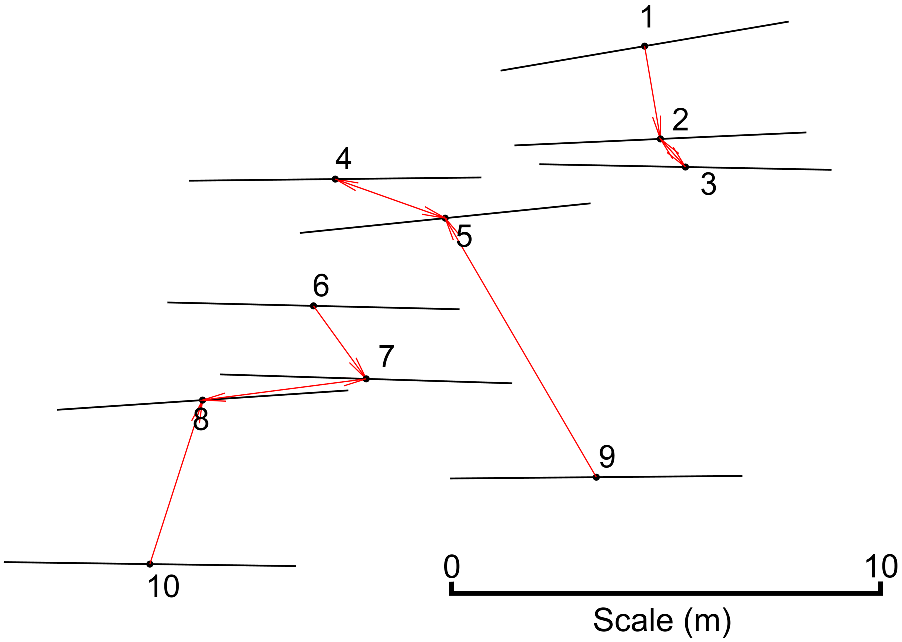


Figure 5.8: The same 10 fractures as in Figure 5.7, but now the bandwidth is reduced to 6 m.

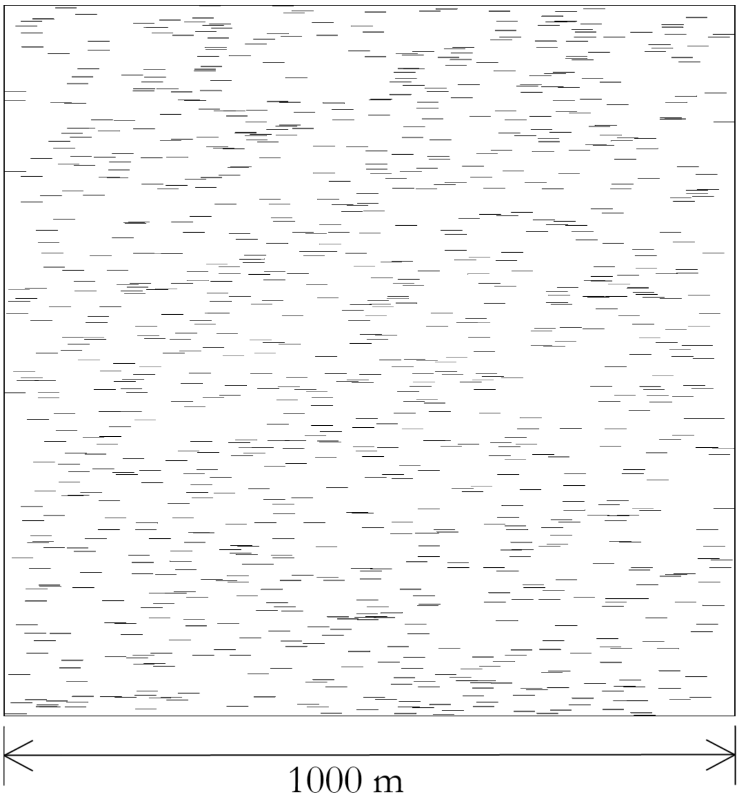


Figure 5.9: DFN with 1000 fractures. All fractures are 30 m long.

5.4.2 Using the Perpendicular Distance for Fracture Spacing

Figure 5.9 shows a DFN with 1000 fractures. The model size is 1000 x 1000 m and all fractures have the same orientation and are the same size (30 m in length). A relative histogram of the fracture spacing (using the scan line methodology) was calculated and is shown in Figure 5.10. The mean fracture spacing is 32 m.

For the same fracture network, the average perpendicular distance to the nearest fracture can be calculated as described above. The average perpendicular distance to the nearest fracture depends upon the bandwidth that is chosen. There is an inverse relationship between average perpendicular distance and bandwidth. As the bandwidth increases, the average perpendicular distance decreases. The relationship between bandwidth and mean perpendicular distance to the nearest fracture is shown in Figure 5.11 (for the simulated DFN shown in Figure 5.9). The histogram of average perpendicular distance for the DFN with 1000 fractures, calculated using a bandwidth of 30 m, is shown in Figure 5.12. The mean perpendicular distance to the nearest

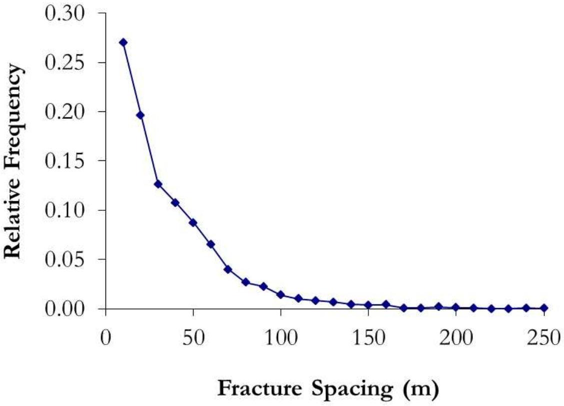


Figure 5.10: Relative histogram of fracture spacing for a DFN with 1000 fractures.

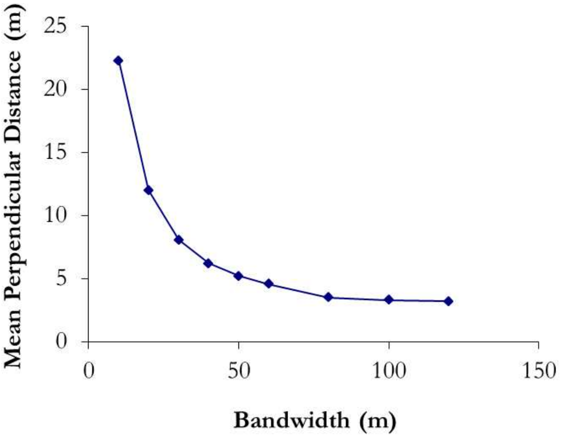


Figure 5.11: The inverse relationship between bandwidth and mean perpendicular distance to the nearest fracture for a DFN with 1000 fractures.

fracture is 8.1 m. Note, the mean perpendicular distance to the nearest fracture will always be less than the mean fracture spacing.

In the next test, the fracture intensity is varied to see if there is a correlation between the mean fracture spacing for the DFNs and the mean perpendicular distance for the same DFN. Six DFNs were generated with fracture intensities of 500, 600, 700, 800, 900 and 1000 fractures. Fracture size was, again, kept constant at 30 m and all fractures are oriented striking in the east-west direction, as before. Using a bandwidth of 30 m (equal to the fracture length), the correlation between the mean fracture spacing and the mean perpendicular distance to the nearest fracture is 0.9988. This shows that as fracture intensity increases, the average fracture spacing and average perpendicular distance decrease proportionally.

The bandwidth was also varied in an attempt to observe how the correlation between fracture spacing and perpendicular distance varies. Figure 5.13 shows how the correlation between mean fracture spacing and mean perpendicular distance changes with increasing bandwidth. The figure shows that, given a particular fracture intensity, fracture length and orientation, there is a near-perfect direct relationship between mean fracture spacing and mean perpendicular distance for any bandwidth that is chosen. However, it is not clear from the figure if there is an optimal choice for bandwidth, based on the correlation between fracture spacing and perpendicular distance. This shows that for a particular fracture intensity, average spacing and average perpendicular distance vary proportionally regardless of the bandwidth chosen.

5.4.3 Another Alternate Measure of Fracture Spacing: Anisotropic Distance

Another problem with the perpendicular distance and bandwidth concept occurs near corners of a rectangular model. If the bandwidth is too narrow, only a few nearby fractures may be identified. Thus, there is a practical lower limit to the choice of bandwidth.

The use of an anisotropic distance in place of the perpendicular distance was investigated in hopes of overcoming the limitations of the perpendicular distance (i.e. the

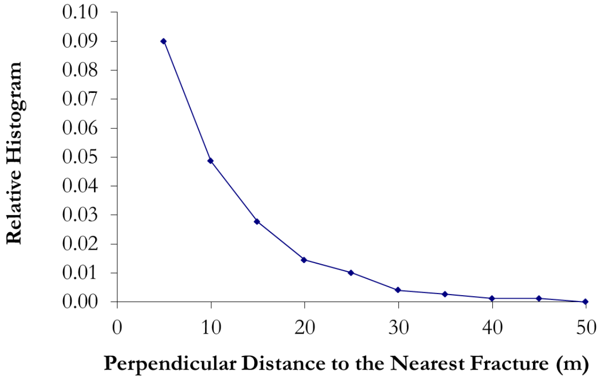


Figure 5.12: Relative histogram of perpendicular distance to the nearest fracture for a DFN with 1000 fractures, using a bandwidth of 30 m.

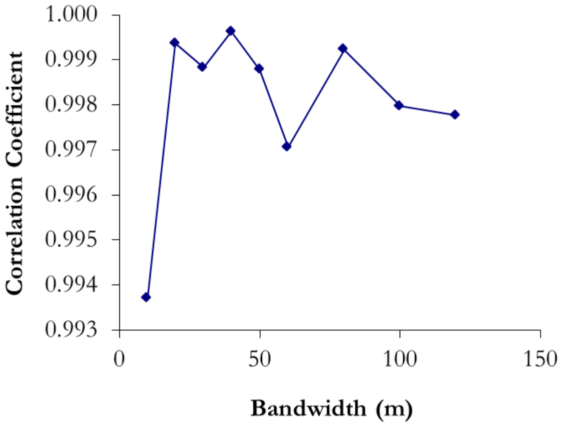


Figure 5.13: The correlation between mean fracture spacing and mean perpendicular distance with increasing bandwidths.

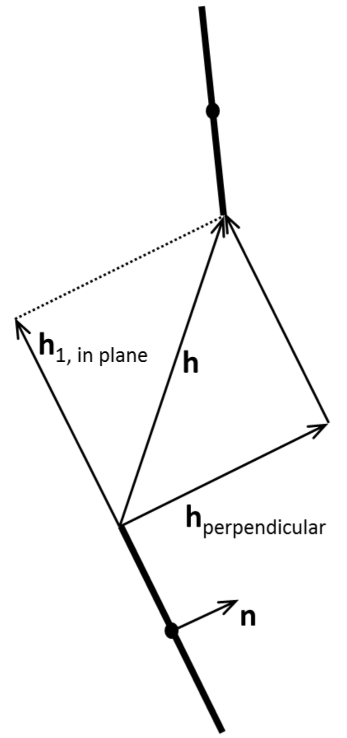


Figure 5.14: Illustration of anisotropic distance. The other component $\mathbf{h}_{2,in-plane}$, is directed out of the page.

hard boundary imposed by the bandwidth). To implement the anisotropic distance, first each fracture is discretized into four points at the corners of the fracture and its centroid. One of the five points is chosen on each of the fractures for the distance calculation. A vector **h** is calculated between the two points. The vector is then resolved into three components. One component is perpendicular to the fracture ($h_{\text{perpendicular}}$), one is horizontal in the plane of the fracture ($h_{\text{horiz, in-plane}}$) and the other is vertical in the plane of the fracture ($h_{\text{vertical, in-plane}}$) (See Figure 5.14).

Next, the three-dimensional anisotropic distance is calculated as an effective distance as follows:

$$h_{anisotropic} = \sqrt{\left(\frac{h_{\text{horiz, in-plane}}}{a_{\text{horiz, in-plane}}}\right)^2 + \left(\frac{h_{\text{vert, in-plane}}}{a_{\text{vert, in-plane}}}\right)^2 + \left(\frac{h_{\text{perpendicular}}}{a_{\text{perpendicular}}}\right)^2} \tag{5.13}$$

where the anisotropy constants, $a_{\text{vert, in-plane}}$, $a_{\text{vert, in-plane}}$ and $a_{\text{horiz, perpendicular}}$, must be specified by the user. Note that the anisotropy constants have units of length, making $h_{anisotropic}$ a unit-less effective distance. For a two-dimensional fracture network, the anisotropic distance does not consider the vertical, in-plane distance:

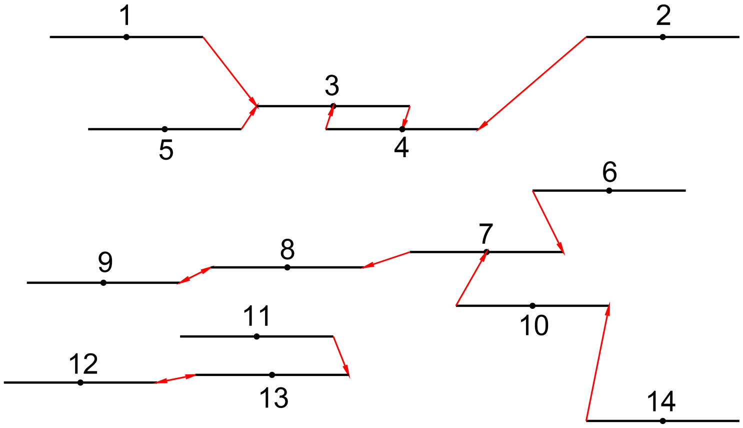


Figure 5.15: Nearest fractures calculated using an anisotropic distance. In this case, $a_{\text{perpendicular}}/a_{\text{in-plane}} = 1$. All fractures are 2 m in length.

$$h_{anisotropic} = \sqrt{\left(\frac{h_{\text{in-plane}}}{a_{\text{in-plane}}}\right)^2 + \left(\frac{h_{\text{perpendicular}}}{a_{\text{perpendicular}}}\right)^2} \tag{5.14}$$

Figure 5.15 shows a two-dimensional fracture network with 14 fractures. The red arrows connect each fracture to their nearest neighbour. In this case, $a_{\text{in-plane}} = a_{\text{perpendicular}}$

$= 1$. The ratio $a_{\text{perpendicular}}/a_{\text{in-plane}} = 1$. Notice that there are several cases of nearest neighbours being identified in a direction that is sub-parallel to the fracture strike. For example, fractures 8 and 9 are identified as nearest to each other, as well as fractures 12 and 13. However, as noted earlier, one would hope to identify nearest fractures in the direction perpendicular to fracture strike and use that distance as a proxy for fracture spacing. In this case, the ratio $a_{\text{perpendicular}}/a_{\text{in-plane}} = 1$ does a poor job of identifying nearest neighbours in the direction perpendicular to the fracture strike since the anisotropic distance is the same as the Euclidean distance between points.

It is possible to give additional weight to fractures in the direction perpendicular to the fracture strike by increasing the $a_{\text{perpendicular}}/a_{\text{in-plane}}$ ratio. Consider Figure 5.16. In this case, $a_{\text{perpendicular}}$ is increased to 2 so that the ratio $a_{\text{perpendicular}}/a_{\text{in-plane}} = 2$. There are several differences between the nearest fractures identified in Figure 5.15 and Figure 5.16. The increase in the anisotropy ratio makes it more likely to identify a nearest neighbour in the direction perpendicular to the fracture strike. When the

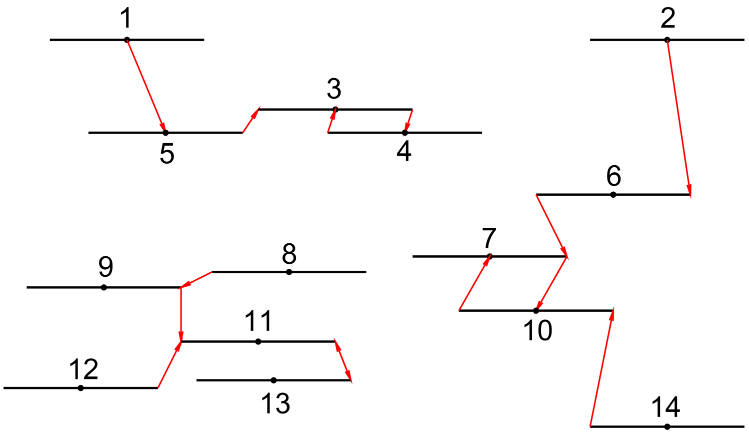


Figure 5.16: Nearest fractures calculated using an anisotropic distance. In this case, $a_{\text{perpendicular}}/a_{\text{in-plane}} = 2$. All fractures are 2 m in length.

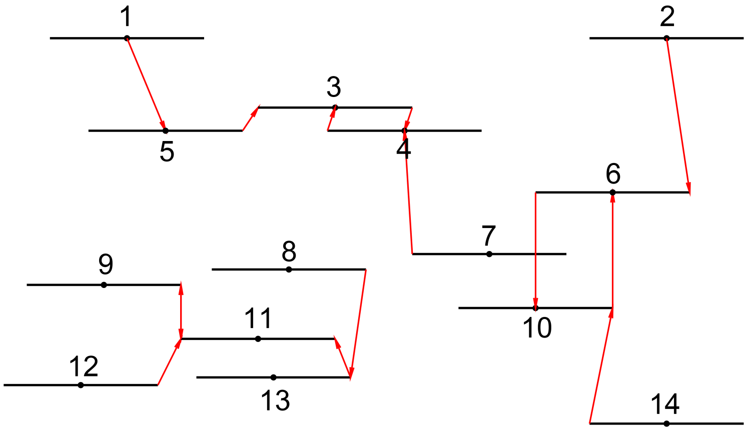


Figure 5.17: Nearest fractures calculated using an anisotropic distance. In this case, $a_{\text{perpendicular}}/a_{\text{in-plane}} = 3$. All fractures are 2 m in length.

ratio was 1, fracture 8 was the closest to fracture 7. When the ratio increases to 2, fracture 10 becomes the new closest fracture to fracture 7. There are several other similar changes to the fracture network. However, there are still some questionable nearest fracture selections with 3 being the nearest to 5 and 9 being the nearest to 8.

There is no change to the result if the $a_{\text{perpendicular}}/a_{\text{in-plane}}$ ratio is further increased to 3. However, if the ratio is increased to 4 the result is Figure 5.17. The figure shows two instances of the algorithm "skipping over" a closer fracture to identify a "nearest" neighbour that is further away. For example, in the figure, fracture 13 is identified as the nearest to fracture 8, ignoring fracture 11, which is positioned between them. Similarly, fracture 10 is identified as the nearest to fracture 6, ignoring fracture 7, which

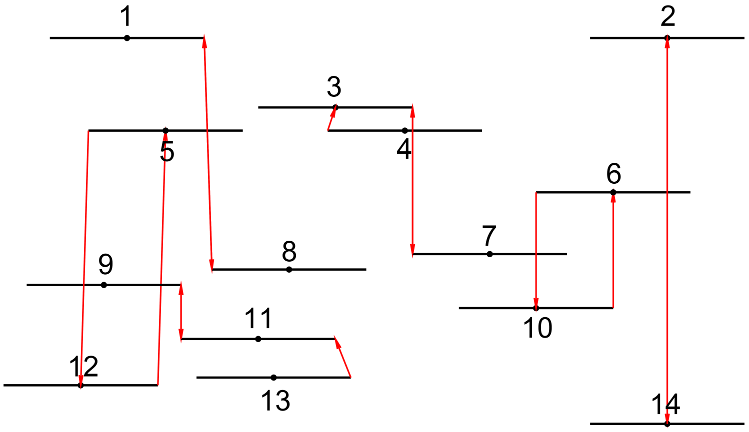


Figure 5.18: Nearest fractures calculated using an anisotropic distance. In this case, $a_{\text{perpendicular}}/a_{\text{in-plane}} = 20$. All fractures are 2 m in length.

is positioned between them.

If the $a_{\text{perpendicular}}/a_{\text{in-plane}}$ ratio is further increased to 20 (see Figure 5.18), the problem of skipping over fractures to identify further away fractures as being the nearest is taken to the extreme. This problem could be minimized by further discretizing the edges of the fractures into more points at the cost of computation speed. The computation time for the anisotropic distance increases with n^2, where n is the number of discretized points on each of the fractures.

If the perpendicular distance calculation is used to calculate the nearest neighbours for the same fracture network, the result is shown in Figure 5.19. For this example, the bandwidth is 2 m, which is equal to the fracture length. The result appears reasonable given that there are no instances of skipping over a fracture to select a nearest fracture that is further away than some other fracture. Moreover, there are no situations where nearest neighbours are selected based on a small in-plane distance.

5.4.4 The Choice Between Perpendicular and Anisotropic Distance

There are merits to both methods of calculating nearest neighbours. The main drawback of the perpendicular distance method is that if the bandwidth is too wide, small perpendicular distances can be calculated for fractures that are far apart in the in-plane direction (along the fracture strike). However, if the bandwidth is too narrow, there may not be enough fractures within the chosen bandwidth. This effect occurs most

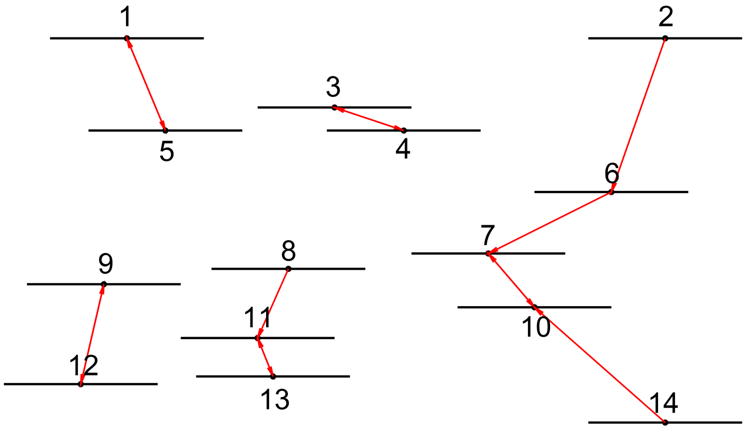


Figure 5.19: Nearest fractures calculated using a perpendicular distance and a 2 m bandwidth. All fractures are 2 m in length.

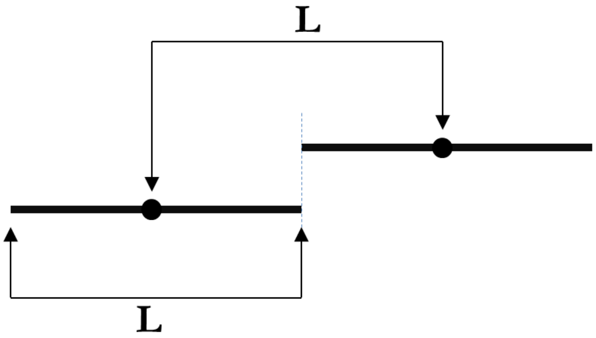


Figure 5.20: When the in-plane distance between centroids is greater than the fracture length, there ceases to be any overlap between the fractures in the perpendicular direction.

often along the edges or near corners of the DFN models. In this way, the anisotropic distance calculation is more robust than the perpendicular distance. The lack of a hard bandwidth boundary removes the problem of finding enough fractures near corners and edges. However, the anisotropy ratio, in the context of fracture spacing, has little physical meaning. It is not clear what an appropriate ratio might be, except through judgement and trial and error.

On the other hand, the choice of bandwidth in the perpendicular distance method is somewhat clearer. One could choose a bandwidth that is related to the average fracture size. For example, a bandwidth that is approximately equal to the mean fracture length seems a sensible choice since, on average, that defines the limit at which fractures cease to overlap in the perpendicular direction (See Figure 5.20).

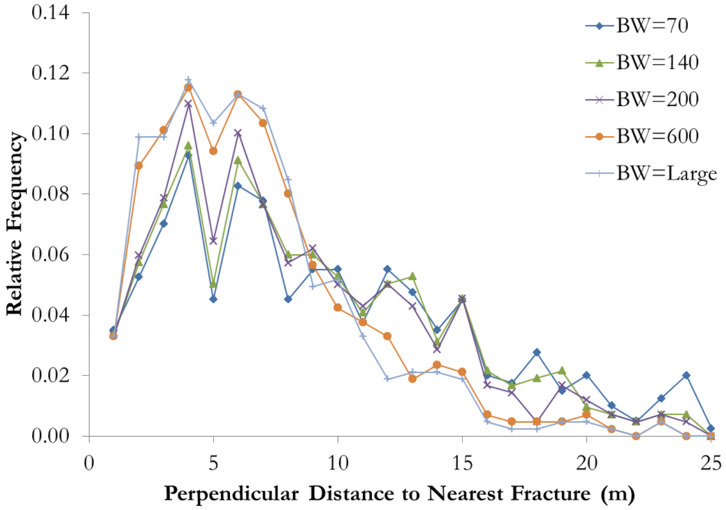


Figure 5.21: Histograms of local fracture spacing with increasing bandwidth. A bandwidth of "Large" means larger than the field size (all fractures are considered).

The perpendicular distance, as currently implemented in `DFNSIM`, is faster than the anisotropic distance calculation. In the anisotropic distance method each fracture is discretized into five points resulting in 25 distance calculations for each pair of fractures versus two in the perpendicular distance method.

It is the author's opinion that the perpendicular distance is also a more intuitive measure of fracture spacing than the anisotropic distance. The perpendicular distance directly measures distance perpendicular to the fracture rather than calculating an off-perpendicular effective distance. The perpendicular distance is also a physical distance (with units) where the anisotropic distance is a unit-less effective distance.

One important consideration in the choice between the perpendicular distance and the anisotropic distance is the sensitivity of the results to the parameter choice (bandwidth for the perpendicular distance vs. anisotropy constants for the anisotropic distance).

A DFN with 850 fractures was created with the intention of observing the impact of varying the bandwidth for the perpendicular distance calculation. The horizontal extents of the DFN are 1000 m in the x-direction and 1050 m in the y-direction. The fractures in the DFN have lengths that follow a normal distribution with a mean of 76 m and a standard deviation of 51 m. Figure 5.21 shows the histograms of perpendicular

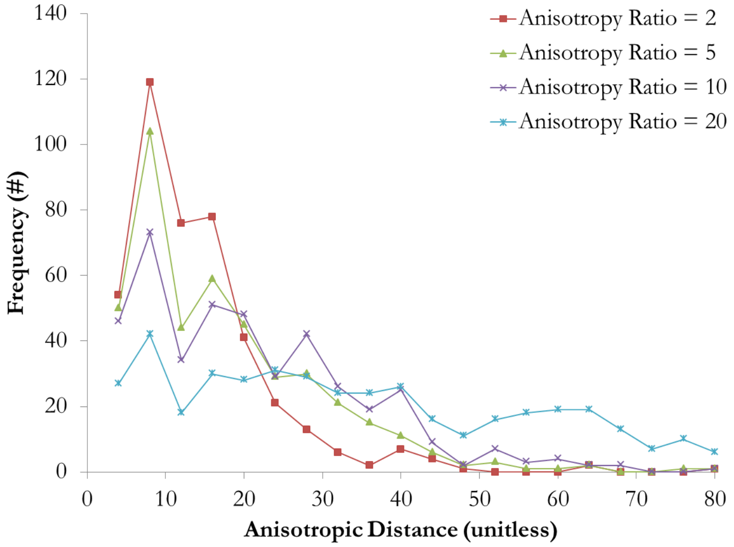


Figure 5.22: Histograms of Euclidean distance to the nearest fracture, where the nearest fracture was determined with anisotropic distance. This chart shows the effect of increasing the anisotropy ratio on the histogram shape.

distance to the nearest neighbour for varying bandwidths. The figure shows that as the bandwidth increases from 70 to 140 and even up to 200 m, there is little change in the histogram of perpendicular distance to the nearest neighbour. When the bandwidth is increased to 600 m (which is larger than half the field size), the histogram changes significantly. In this example, the average fracture size is 76 m, so it seems that as long as the bandwidth is between one and three times the average fracture size, there is little change to the histogram of perpendicular distance to the nearest neighbour.

Ideally, one would also construct histograms of anisotropic distance for changing values of the anisotropy ratio ($a_{\text{perpendicular}}/a_{\text{in-plane}}$), however this comparison is meaningless since the anisotropic distances are scaled by the constants $a_{\text{perpendicular}}$ and $a_{\text{in-plane}}$. This means that the effective anisotropic distances to the nearest neighbour get smaller with increasing $a_{\text{perpendicular}}$, even if the same fractures are chosen as the nearest neighbours. Thus, Figure 5.22 shows relative histograms of Euclidean distance to the nearest fracture, where the nearest fractures are determined by the anisotropic distance. Four relative histograms are shown for four different anisotropy ratios.

Figure 5.22 shows that as the anisotropic ratio $a_{\text{perpendicular}}/a_{\text{in-plane}}$, increases, there's a big increase in the width of the histograms. This means that the distances to the nearest neighbours increases as the ratio increases. Although the histograms in Figures 5.21 and 5.22 are not directly comparable, based on visual inspection, it appears that the anisotropic distance measure is more sensitive to its parameter ($a_{\text{perpendicular}}/a_{\text{in-plane}}$ ratio) than the perpendicular distance is to its parameter (bandwidth).

5.5 How to Build Input Histograms

In addition to using `DFNSIM` to simulate DFNs, it can also be used to build the target histograms of local fracture spacing and orientation. If a natural fracture network can be digitized (say from a photo of an outcrop), those fractures can be read in by `DFNSIM` and the program will build the histograms of local fracture spacing, local fracture orientation, and fracture length and can also calculate the number of intersections. The steps that the program takes to calculate the histograms have already been described (See Section 4.4). Recall the first few steps of the `DFNSIM` algorithm:

1. Randomly simulate a pool of fractures
2. Assign fractures to the *activated DFN*
3. Calculate histograms of local fracture spacing, local fracture orientation and fracture length. Also calculate the number of fracture intersections and fracture intensity of the activated DFN.

Target histograms can be constructed by simply reading in digitized fractures in place of the simulated pool and assigning all of them to the activated DFN before calculating the DFN spatial statistics as in Step three.

5.6 An Example Application of `DFNSIM`

In order to further explain a few key `DFNSIM` parameters, a short example is presented. Section 4.3.1 introduces a map of lineaments from Northern Alberta inferred from satellite imagery and digital elevation models (also see Figure 4.2). The Southwest-Northeast lineament set was digitized. In total, 425 lineaments were digitized.

The lineaments have an average pole trend of 146.5 degrees, clockwise from North with a standard deviation of 11.8 degrees. Although the rose diagrams indicate that the lineaments are from a single set (Figure 4.3), from a qualitative standpoint there appears to be some pattern to the lineament locations and orientations. Namely, there are relatively few (20) lineament intersections and lineaments generally appear to be oriented similarly to their nearest neighbours. While there is variation in the orientation of the lineaments when looking at an area wide scale, locally lineaments appear to be oriented similarly to their nearest neighbours.

`DFNSIM` was used to simulate a model of the lineaments. Then the spatial statistics of the simulated lineament model were compared to the original digitized lineaments. `DFNSIM` was used to calculate target input histograms for local lineament spacing (perpendicular distance to the nearest neighbour), deviation in local lineament orientation and lineament length. There are 425 lineaments with 20 intersections on the map of lineaments, which are the target intensity and number of intersections, respectively. Figure 5.23, shows the target histograms of local lineament spacing (i.e. perpendicular distance to the nearest lineament), deviation in local lineament orientation and lineament length that were calculated from the digitized lineaments shown in Figure 4.2. The goal was to generate a DFN that matches these target histograms as well as the target intensity and number of lineament intersections.

The FMF (the fracture multiplication factor) was set at 2. This means that 850 lineaments were generated for the pool and half were assigned to the initial DFN. Thus, there were 425 lineaments in the initial DFN since the proposed algorithm activates the target intensity to begin with. The initial DFN is shown on the left side of Figure 5.24. Visual inspection of the initial DFN shows many instances of lineaments that are extremely close together compared to the map of digitized lineaments. There are 86 lineament intersections in the initial DFN, which is more than four times as many as the target of 20. Figure 5.25 shows the target, initial DFN and final DFN histograms of local lineament spacing, deviation in local lineament orientation and lineament length. Note that the initial DFN histograms of perpendicular distance to the nearest lineament and deviation in local lineament orientation are poor matches for the target histograms.

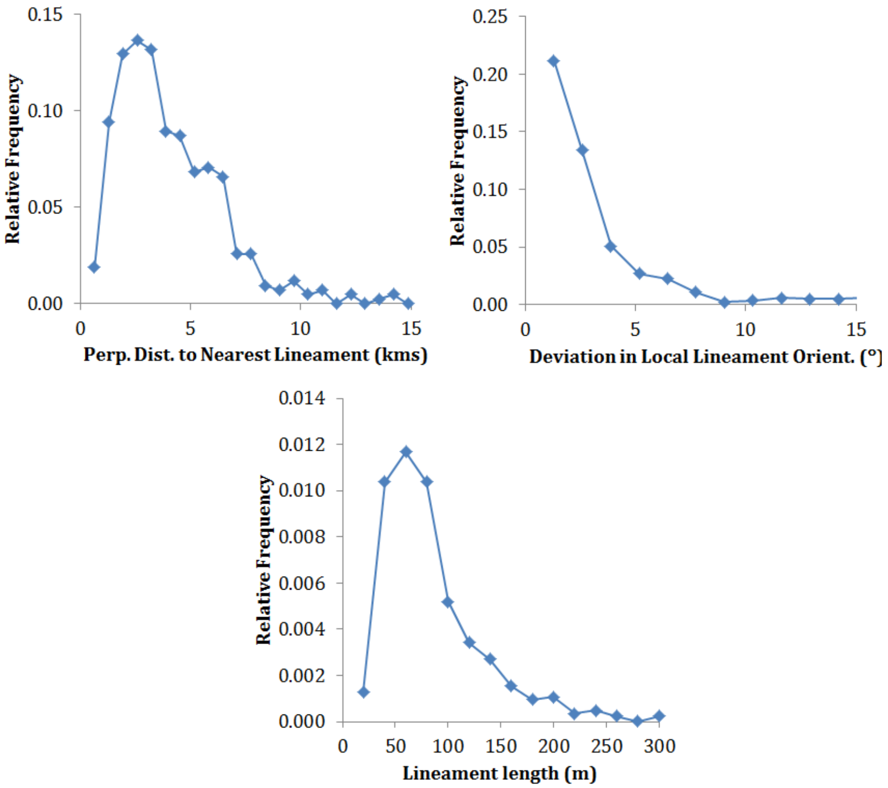


Figure 5.23: Target relative histograms of perpendicular distance to the nearest lineament, deviation in local lineament orientation and lineament length.

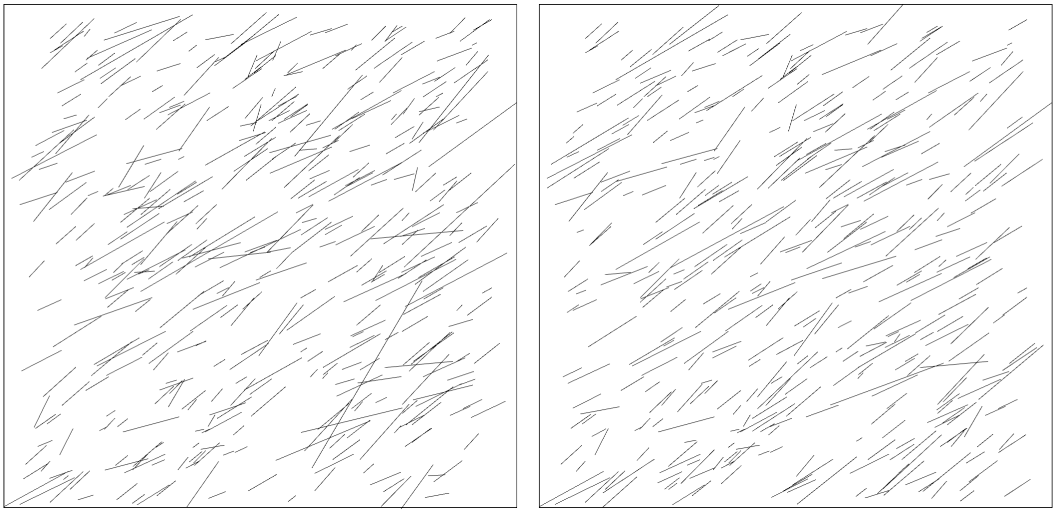


Figure 5.24: On the left: An initial DFN generated to represent the lineaments on the map of lineaments. On the right: The optimized DFN after 3400 iterations (visiting each of the 850 lineaments 4 times). 1 cm on the figure represents 8.2 km.

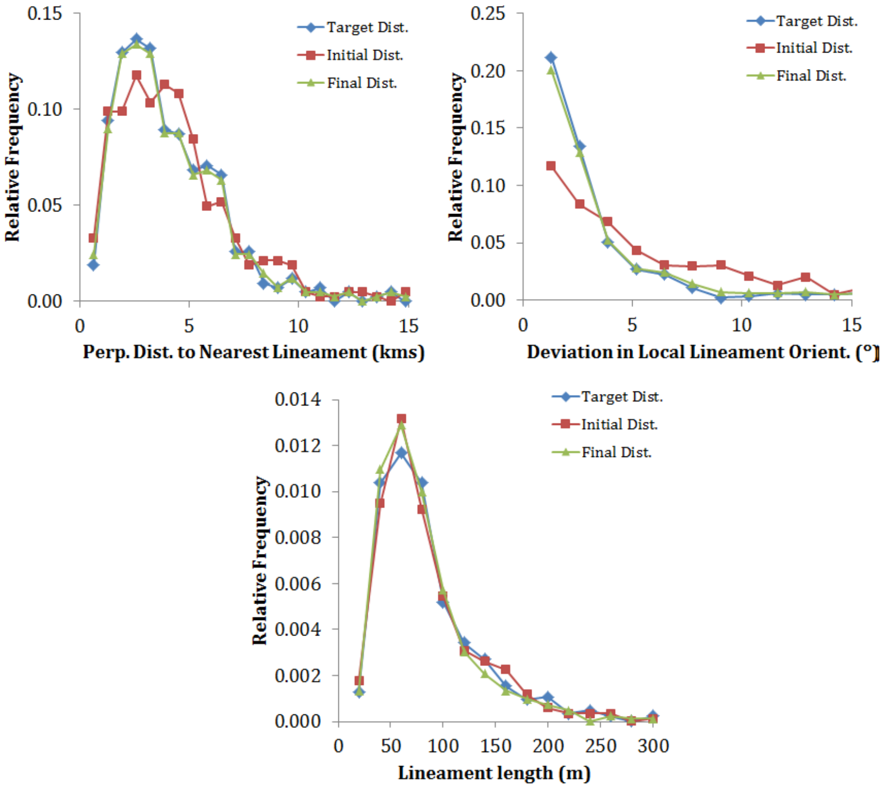


Figure 5.25: Target, initial DFN and Final (optimized) DFN relative histograms of perpendicular distance to the nearest lineament, deviation in local lineament orientation and lineament length.

This was expected given the conclusions reached in Chapter 4. However, the initial DFN is a good match in terms of lineament length. Lineament length was drawn from a log-normal distribution with a mean and standard deviation matching the input data. As it turns out, the histogram of lineament length closely matched the log-normal distribution.

The DFN was optimized using four iteration loops, which means that each of the 850 lineaments were visited four times. This means that there the DFN was subject to 3400 changes (iterations). The objective function result for the initial DFN was 1.6 x 10^6 while the objective function result for the final DFN is 3.1.

Visual inspection of the optimized DFN (right side of Figure 5.24) shows that the lineaments are oriented more similarly to their nearest neighbours and have a more regular spacing, as is evidenced by the fewer bare spots (areas without lineaments), compared to the initial DFN.

Figure 5.25 also shows the final (optimized) DFN histograms of local lineament spacing, deviation in local lineament orientation and lineament length. All three histograms for the optimized DFN are a very good match to the target histograms. The final optimized DFN had 413 lineaments (a 2.8% margin of error) and 20 intersections, which exactly matched the target number of intersections.

This example has shown 1) how to generate target fracture network statistics and 2) how the proposed DFN simulation algorithm can be used to generate DFNs that honour the target spatial statistics.

5.6.1 Choice of FMF

In the DFN simulation example from the previous section, the FMF was set to 2, which means the pool of lineaments is twice as large as the target intensity. It is difficult to recommend a precise number for the FMF. However, as the FMF increases, the objective function result decreases, meaning the match between the target statistics and the DFN's statistics improves.

Figure 5.26 shows how changing the FMF affects the objective function result for the example problem. Increasing the FMF from two to three results in a substantial improvement in the objective function result. For this case, the spatial statistics of the final DFN match the targets even better than before. Figure 5.27 shows the target, initial DFN and final DFN histograms for the case where the FMF equals three. There is a noticeable improvement in the match between the target and final DFN histograms compared to the case when the FMF was two (see Figure 5.25). The number of lineaments has increased to 427, which decreases its error from the target from 2.8% to 0.5%.

Figure 5.26 shows that, in this case, increasing the FMF beyond three does not improve the result, in terms of the objective function result. While testing `DFNSIM`, the author has never found a case where the FMF needed to be greater than three in order to obtain an acceptable optimization. Thus, it is recommended to use a FMF of three. If the convergence results are unacceptable, the FMF or the number of iteration loops can be increased.

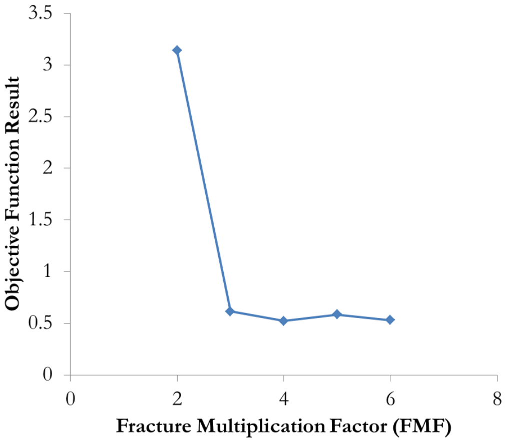


Figure 5.26: The affect of FMF on the objective function result. Note, when the FMF is one, the objective function equals 284 and there is no optimization since all fractures must be part of the DFN.

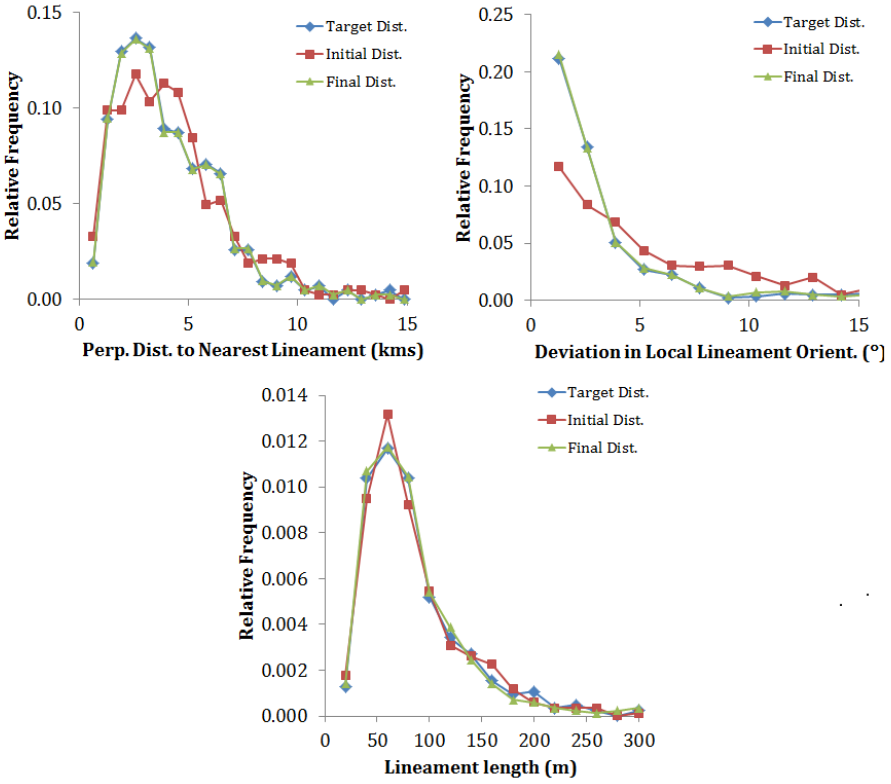


Figure 5.27: Target, initial DFN and final (optimized) DFN relative histograms of perpendicular distance to the nearest lineament, deviation in local lineament orientation and lineament length. In this case, the FMF is 3.

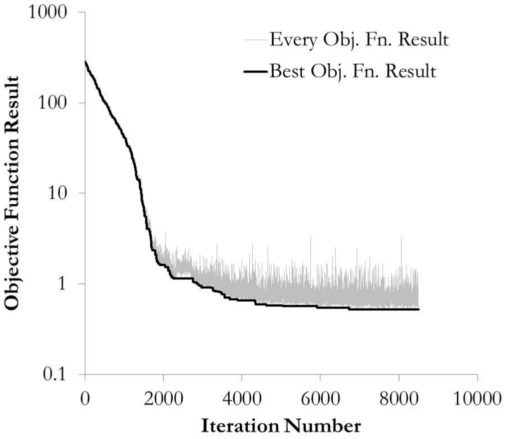


Figure 5.28: The objective function result as a function of iteration number. The result of every objective function calculation is shown in the thin grey line. The current lowest objective function result is shown in the thick black line. In this case, the FMF is four and the number of iteration loops is five.

5.6.2 Convergence

Figure 5.28 shows the progress of the objective function over the course of the optimization. The figure shows that the objective function decreases fastest in its first 2000 iterations. After iteration number 3000, very little improvement is made. In this example, the FMF is four, so there are 1700 lineaments in the pool (recall that the target number is 425). 3400 iterations means that each lineament in the pool is visited twice during the optimization. Thus, in this case, almost all of the improvement to the DFN is made in the first two iteration loops. In fact, while testing `DFNSIM`, the author has not come across any examples where more than three iteration loops were required.

5.6.3 Stopping Criteria

A stopping criteria determines when an optimization process ends. The proposed algorithm uses one of the most popular stopping criterions, which is simply to stop the optimization process after a certain number of objective function evaluations (Zielinski and Laur, 2005). The advantage of this stopping criteria is that it is extremely simple to implement. The disadvantage is that an appropriate number of evaluations is hard to define. The appropriate number of objective function evaluations is dependent on the optimization problem and is subject to fluctuations due to the randomness of the algorithm and is usually determined by trial and error (Zielinski and Laur, 2005).

Alternative stopping criterion could have been considered. One could stop the optimization when the objective function reaches a desired result; however, defining an appropriate target result is difficult since the objective function is unit-less and mixes multiple components. One could use a stopping criteria based on computation time; however, the computation time in this case depends largely on the size of the fracture network and is analogous to stopping after a certain number of iterations. A good idea might be to stop the optimization if objective function improvements slow down. That is, the optimization could stop if an improvement has not been seen in a certain number of iterations, or if the objective function gradient decreases below a certain tolerance.

5.7 Reproducing the True Fracture Spacing Histogram

For reasons stated earlier, the proposed fracture simulation algorithm uses the perpendicular distance to the nearest fracture as an alternative measure of fracture spacing. Figure 5.25 shows the good match between the target and final DFN histograms of perpendicular distance to the nearest fracture. However, it is also of interest to compare the fracture spacing histograms (using the scan line method) of the final optimized DFN and the digitized fractures. Figure 5.29 shows the histogram of fracture spacing for the digitized lineaments, the initial DFN and the final optimized DFN. The figure shows that the initial DFN is a poor match for the digitized fractures in terms of the true fracture spacing histogram. The final DFN shows only a slight improvement and is generally still a very poor match for the spacing of the digitized fractures.

5.7.1 Using the Perpendicular Distance as an Alternative Measure of Fracture Spacing

An investigation was undertaken to try to discern a reason for the poor match between the target and optimized DFN fracture spacing histograms, despite obtaining very good matches for the perpendicular distance histograms.

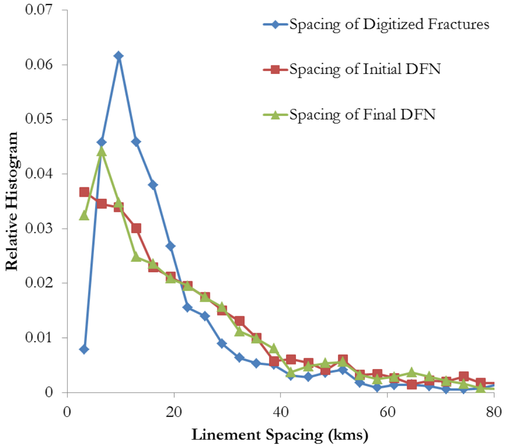


Figure 5.29: True fracture spacing of the digitized fractures, the initial DFN and the final optimized DFN.

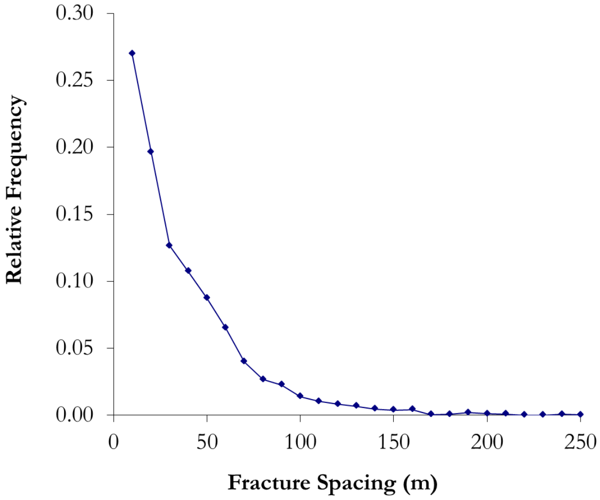


Figure 5.30: Relative histogram of fracture spacing for a DFN with 1000 fractures.

Correlation Between Fracture Spacing and Perpendicular Distance in a DFN

First, a DFN with 1000 fractures was simulated in an area of interest that is 1000 x 1000 m. The DFN was two-dimensional and was simulated using the standard Baecher model for simplicity. All fractures have the same orientation and are 30 m in length. Fracture spacing was calculated along scan lines that are perpendicular to the fracture orientation and a relative histogram of fracture spacing was constructed and is shown in Figure 5.30. The mean fracture spacing is 32 m.

For the same fracture network, the average perpendicular distance to the nearest

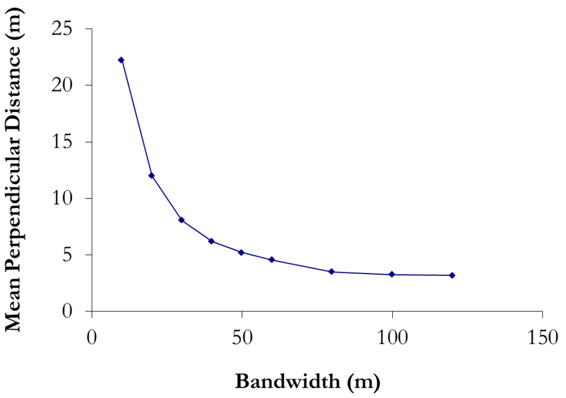


Figure 5.31: The inverse relationship between bandwidth and mean perpendicular distance to the nearest fracture for a DFN with 1000 fractures.

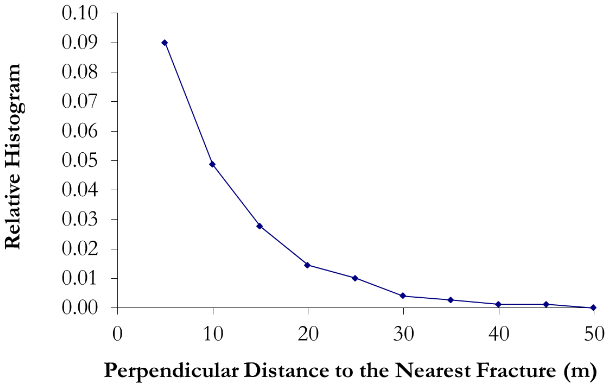


Figure 5.32: Relative histogram of perpendicular distance to the nearest fracture for a DFN with 1000 fractures, using a bandwidth of 30 m.

fracture was calculated. An additional challenge is that the perpendicular distance to the nearest fracture depends upon the choice of bandwidth. There is an inverse relationship between perpendicular distance to the nearest neighbour and bandwidth. As the bandwidth increases, the average perpendicular distance decreases. The relationship between bandwidth and mean perpendicular distance to the nearest fracture for the simulated DFN is shown in Figure 5.31. An example of the histogram of perpendicular distance to the nearest fracture, using a bandwidth of 30 m, is shown in Figure 5.32. The mean perpendicular distance to the nearest fracture is 8.1 m.

Next, one hundred DFN realizations were simulated by using the same settings while varying only the random number seed. For each realization, the average fracture spacing and average perpendicular distance to the nearest neighbour were calculated.

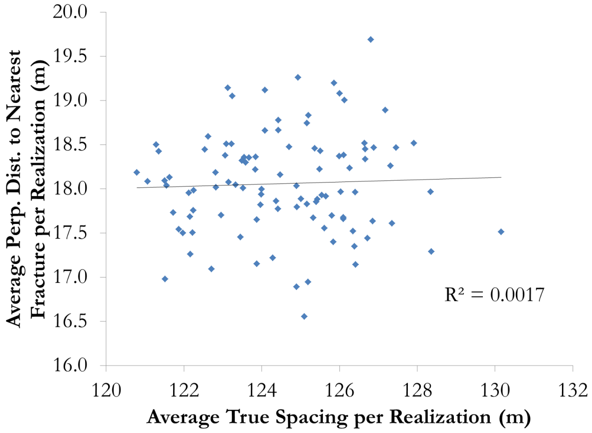


Figure 5.33: The mean fracture spacing versus the mean perpendicular distance for each of 100 DFN realizations.

Figure 5.33 shows a scatter plot of the average fracture spacing versus the average perpendicular distance for each one of the 100 DFN realizations. The scatter plot shows no correlation between the average true fracture spacing and the average perpendicular distance per realization. The effect of increasing and decreasing bandwidth was investigated but was found to make no difference to the results.

5.7.2 Reducing the Problem to One Dimension

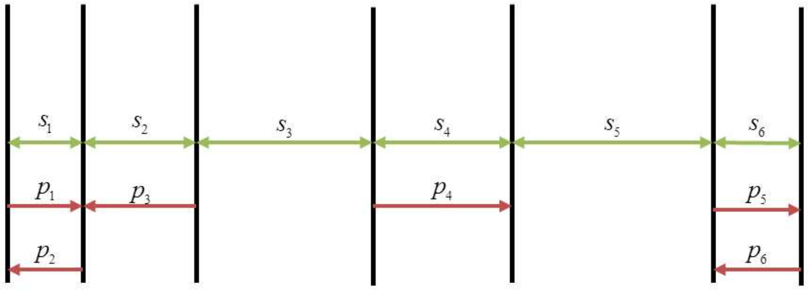
Figure 5.34 shows a depiction of fractures perpendicular to a one-dimensional line at coordinate locations x_i where $i = 1, ..., n$. The intersections between the fractures and the line, which could be seen as a scan line, defines the fracture spacings:

$$s_i = x_i - x_{i-1},\ i = 2, ..., n. \tag{5.15}$$

The perpendicular distances, p_i between fractures are calculated as the minimum of: 1) the distance to the *previous* fracture, or 2) the distance to the *next* fracture. Mathematically that is:

$$p_i = \min(x_i - x_{i-1},\ x_{i+1} - x_i) \tag{5.16}$$

Note that the perpendicular distance p_i is always less than or equal to s_i.



s_i = Fracture spacings
p_i = Perpendicular distances to nearest fractures

Figure 5.34: 11 one-dimensional fractures, simulated along a line.

For this example, 10,000 fractures were simulated randomly along a one-dimensional line, and then were sorted by x-coordinate location. The sorted fracture centroid locations allow calculation of spacing between them as in Equation 5.15. The perpendicular distances between each fracture and its nearest neighbour are also calculated as in Equation 5.16. This process was repeated 20 times, resulting in 20 realizations of fractures. The mean spacing and mean perpendicular distance to the nearest neighbour were calculated for each realization. Figure 5.35 shows the relationship between the mean values of true spacing and perpendicular distance for each of the 20 realizations. There is zero correlation between the average true fracture spacing per realization and the average perpendicular distance to the nearest neighbour.

Next, the individual distances and spacings, rather than the average spacing and perpendicular distances, were examined. Figure 5.36 shows the relationship between individual values of perpendicular distance to the nearest fracture and fracture spacing for one of the 20 realizations (this is a scatter plot of p_i vs. s_i where $i = 2, ..., n$). The correlation between p_i and s_i is 0.5. While a correlation of 0.5 is significant, this case only considers one dimension (fractures on a line). The relationship shown in Figure 5.36 would have a lower correlation in two or three dimensions.

Perpendicular Distance to Nearest Fractures in Opposite Directions

It was thought that there might be an improvement in the correlation of fracture spacings and perpendicular distances by considering an alternative measure of perpen-

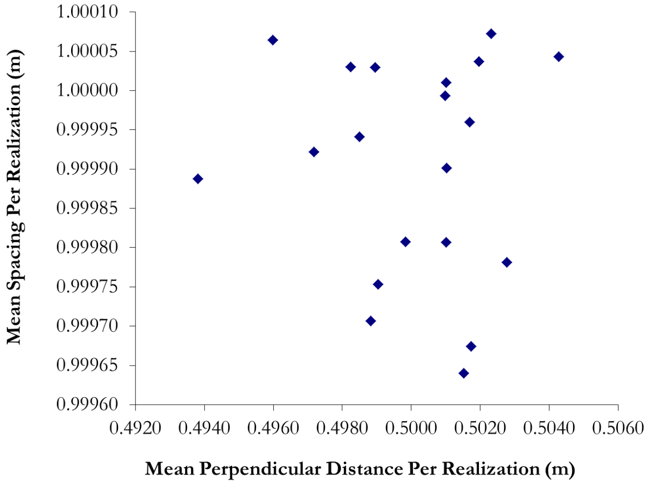


Figure 5.35: Relationship between mean spacing and mean perpendicular distance per realization.

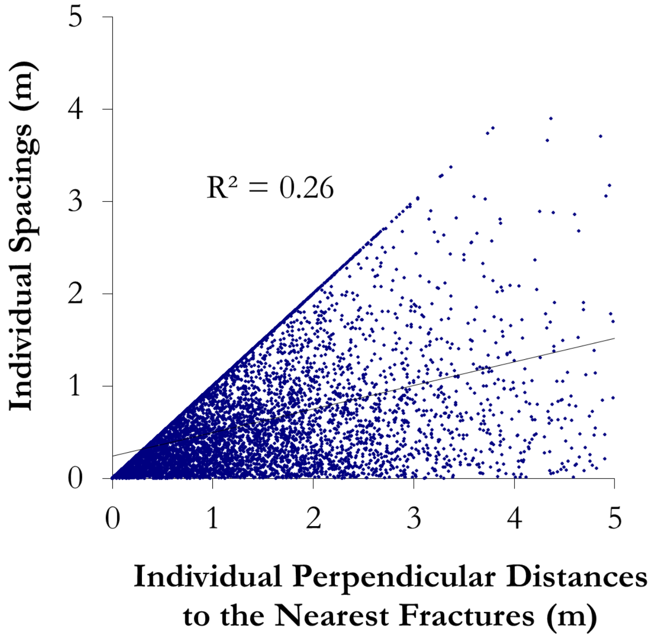
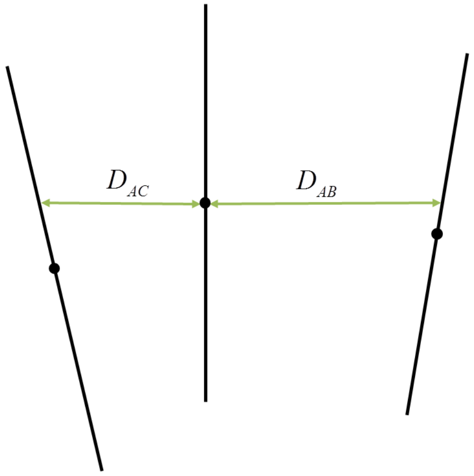


Figure 5.36: Relationship between individual values of perpendicular distance and spacing.



$$\text{Perp. Dist. from A to Nearest Neighbours} = D_{AB} = \frac{D_{AC} + D_{AB}}{2}$$

Figure 5.37: New method for calculating the perpendicular distance to the nearest neighbours. The perpendicular distance is the average of the two distances in opposite directions.

dicular distance. For this next study, the average of two perpendicular distances *in opposite directions* was considered. Figure 5.37 shows the calculation of the modified perpendicular distance that considers opposite directions. If the fractures occur along a one-dimensional line, this amounts to:

$$D_i = \text{average}(x_i - x_{i-1}, x_{i+1} - x_i) = \frac{[x_i - x_{i-1}] + [x_{i+1} - x_i]}{2}, \; i = 1, ..., 1000 \quad (5.17)$$

A simulation study was conducted where 1000 fractures, f_i, are simulated along a one-dimensional line (f_i where $i = 1, ..., 1000$). Fracture locations and orientations are generated according the following rules:

- $x_1 = 0.5$
- a new fracture location, x_i, is generated by adding the previous value (x_{i-1}) to a value randomly selected from a normal distribution with mean = 1 and standard deviation = 0.5.
- all fractures are oriented perpendicular to the line

The idea in this study is to simulate fractures on a one-dimensional line that have

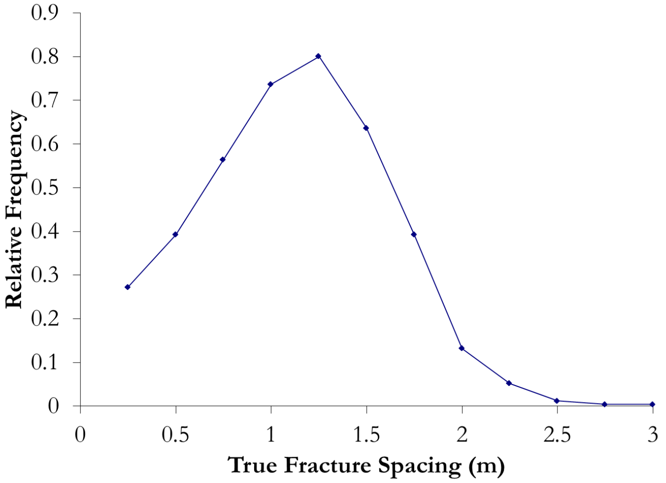


Figure 5.38: Relative histogram of true fracture spacing.

semi-regular spacing. That is, the average spacing should be 1 m. This is indicated on the histogram of fracture spacing, which is shown in Figure 5.38. Next, the perpendicular distances that consider both directions (Equation 5.17) are calculated. Figure 5.39 shows the scatter plot of fracture spacings versus the modified perpendicular distances considering opposite directions. The correlation coefficient is 0.71, which indicates an improvement over the case where the perpendicular distance only considers one direction.

The 1000 pseudo-random fractures are used as the “Truth” and a modified version `DFNSIM` is used to model those fractures. The target histogram of average perpendicular distribution in opposite directions is built by the modified version of `DFNSIM`. 1000 fractures are simulated with the target histogram of average perpendicular distance in opposite directions. The target, initial DFN and final DFN relative histograms of average perpendicular distance in opposite directions are shown in Figure 5.40. Note the excellent match between the target and final DFNs.

Figure 5.41 shows the correlation between fracture spacing and average perpendicular distance considering opposite directions for the initial DFN ($r = 0.707$). Figure 5.42 shows the correlation for the final optimized DFN ($r = 0.57$). The process of optimizing fracture locations to match the target histogram of perpendicular distance results in a lower correlation to the fracture spacings.

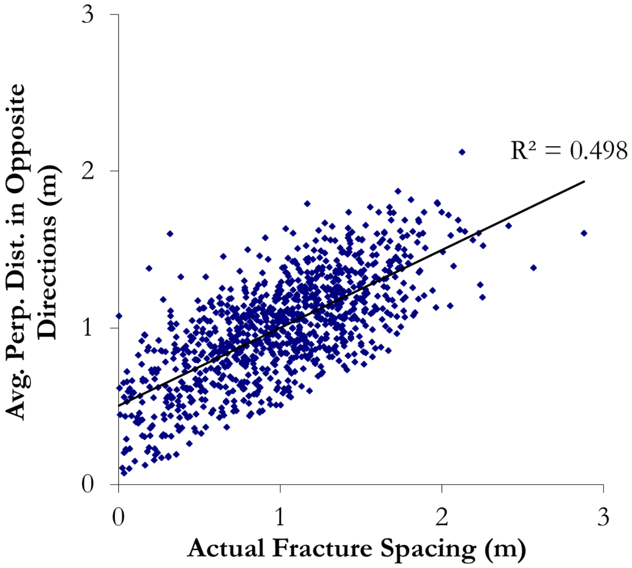


Figure 5.39: Correlation between true fracture spacing and average perpendicular distance in opposite directions.

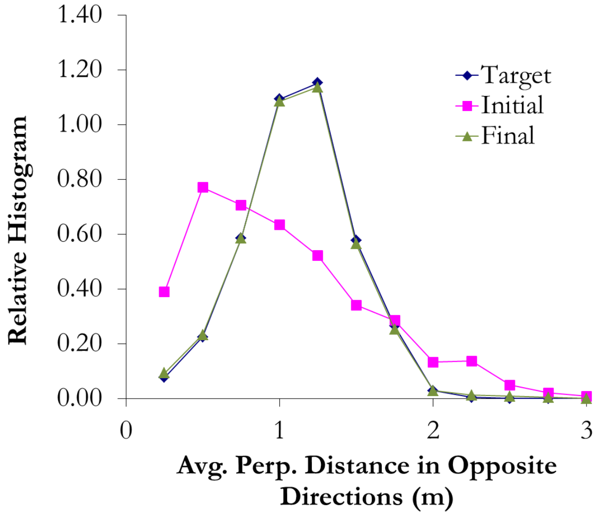


Figure 5.40: Target, initial DFN and final DFN histograms of average perpendicular distance in opposite directions. Note that the target and final data nearly overlay each other.

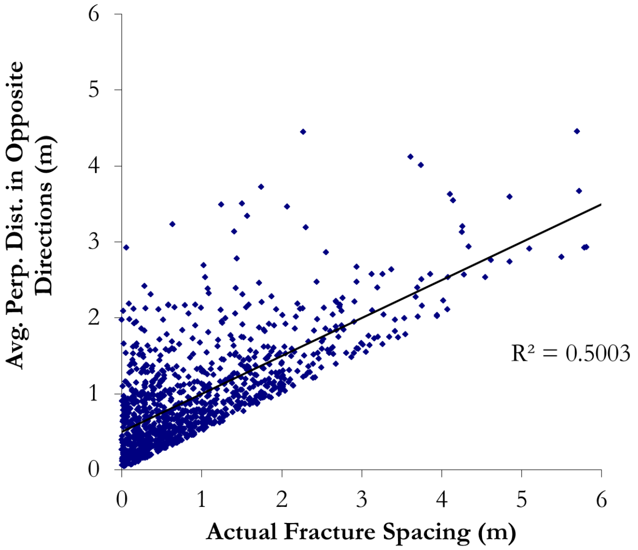


Figure 5.41: True fracture spacing versus average perpendicular distance in opposite directions for the initial DFN.

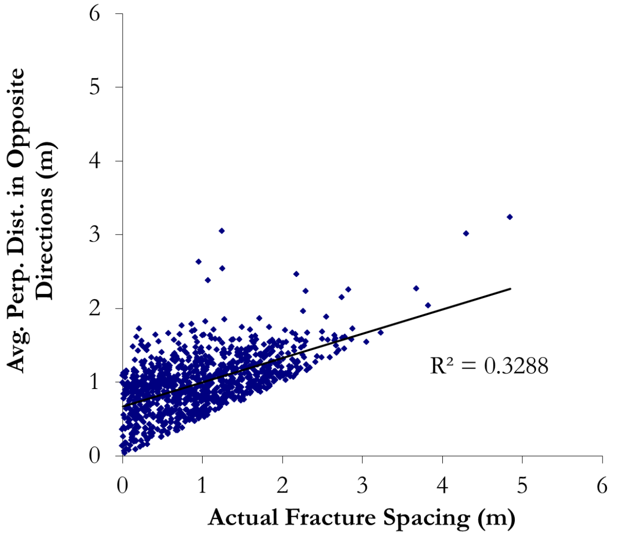


Figure 5.42: True fracture spacing versus average perpendicular distance in opposite directions for the final DFN.

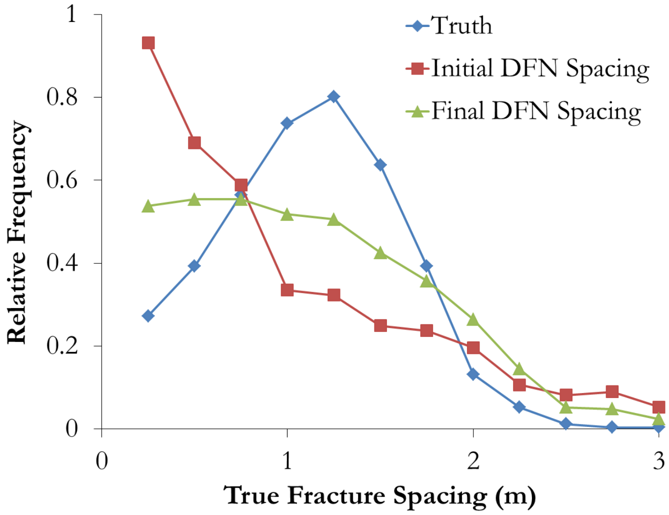


Figure 5.43: True fracture spacing for the original fractures (i.e. the truth), the initial DFN and the final DFN.

Finally, consider Figure 5.43, which shows the fracture spacing histogram of the DFN that is considered the “truth”. The histograms of fracture spacing for DFN intended to model those fractures are also shown for the initial DFN and the optimized DFN. The fracture spacing histogram for the optimized DFN is an improvement over that of the initial DFN; however, the histogram of fracture spacing for the optimized DFN is a poor match for the histogram of spacing for the truth DFN.

Remarks on Spacing Validation Studies

The proposed fracture simulation algorithm uses perpendicular distance to the nearest fracture as an alternative measure of fracture spacing. An alternative measure of fracture spacing was implemented because the traditional way to measure fracture spacing proved too computationally expensive when iterating through large DFNs with millions of fractures. Results presented in this thesis show that it is possible to optimize DFNs to honour a target histogram of perpendicular distance to the nearest fracture. Unfortunately, the connection between fracture spacing and perpendicular distance is weak, even in one dimension. The result is that the optimized DFNs will not honour a desired distribution of fracture spacing in the traditional sense, even if a good match is obtained for perpendicular distance to the nearest fracture. The perpendicular distance considering opposite directions was considered; but the improvement it provided was minimal.

In the future, it is recommended that the fracture simulation algorithm be re-worked to optimize on the more traditional fracture spacing, rather than the perpendicular distance to the nearest fracture. Although this proved computationally expensive, there may be some means of optimizing the calculations to improve speed to an acceptable level. This is discussed in more detail in the final chapter (See Section 7.3: Future Areas of Research).

5.8 `DFNSIM` Parameter File

Table 5.1 shows the `DFNSIM` parameter file. Line numbers are provided for reference purposes. Some parameters are self explanatory and some require more explanation, which is provided below.

Table 5.1: The `DFNSIM` Parameter File.

```
 1                           Parameters for DFNSIM
 2                           *********************
 3
 4 START OF PARAMETERS:
 5 1  5 10             -nx,xmn,xsiz
 6 1  5 5              -ny,ymn,ysiz
 7 1  0.05 0.1         -nz,zmn,zsiz
 8 6939                -random number seed
 9 inputdata.dat       -input fracture data (honoured in final models)
10 ----------------------------------------------------------------------
11 ***DATA SEARCH PARAMETERS***
12 20000 20000 1000   -maximum search radii
13 -1                 -number of nearest data to perp. search, <0 for all
14 2                  -Band width for perpendicular fracture search
15 40000              -Perpendicular distance to search
16 ----------------------------------------------------------------------
17 ***OUTPUT FRACTURE FILES***
18 fracdata.out           -Output GSLIB-style fracture data
19 fracdata.fab           -Output FracMan fracture data
20 fracdata_optimized.out -Optimized output GSLIB-style fracture data
21 fracdata_optimized.fab -Optimized output FracMan fracture data
22 ----------------------------------------------------------------------
23 ***INPUT FRACTURE DISTRIBUTIONS***
24 spacingdist.out     -Target Fracture Spacing Distribution
25 angledist.out       -Target Inter-Fracture Angle Distribution
26 lengthdist.out      -Target Fracture Length Distribution
27 intensity.out       -Target Fracture Intensity Input File
28 1  1               -Int. real. to use, # of realizations in file
29 1                  -1 = Fracture Count, 2 = P32 (frac area/volume)
30 ----------------------------------------------------------------------
31 ***FRACTURE SET PARAMETERS***
```

```
32 1                -Load initial DFN from file? (1=yes, 0=no)
33 initialDFN.out   -File with Initial DFN
34 0                -Use locally varying Fracture orientation (1=yes,0=no)
35 lvo.out          -If "Yes", specify file
36 1                -Fracture orientation distribution (see below)
37 0.0 0.0          -Avg. pole trend and plunge
38 2 0.01           -St. dev. of pole trend and plunge / dispersion
39 1                -Fracture length and height dist. (see below)
40 1.5 0.01 1.5     -Fracture length mean, st. dev., minimum value
41 1                -Fracture height distribution (see below)
42 0.01 0.02 0.02   -Fracture height mean, st. dev., minimum value
43 2                -Fracture aperture distribution (see below)
44 7.2 0.77 0.01    -Fracture aperture mean, st. dev., minimum value
45 0                -Random z location for fractures? (1=yes,0=no)
46 2                -Fracture Intensity Multiplication Factor
47 ----------------------------------------------------------------------
48 ***OPTIMIZATION PARAMETERS***
49 0                -Optimize? 1=yes,0=no just output initial dfn.
50 4                -Number of iteration loops
51 1 1 1 1 1        -Optimize on spacing, orientation, length,
52                        intensity, intersections?
53 1 1 1 .1 0.1     -Scaling Constants for components of obj. fn.
54 20               -Target Number of Intersections
55 ----------------------------------------------------------------------
56 ***OTHER PARAMETERS***
57 1                -output objective function progress? (1=yes, 0=no)
58 iterations.out   -File for iteration objective function progress
59 results.out      -File for spacing and interangle results
60 ---------------------------------------------------------
61 Distributions to draw from (these lines are not read in):
62 1 = Normal (specify mean and st dev)
63 2 = Lognormal (specify mean and st dev)
64 3 = Fisher (specify dispersion)
65 4 = Parametric Distribution (specify file with file for parametric
66     distribution on next line instead of distribution parameters)
```

5.8.1 Fracture Set Parameters

Line 35 in the `DFNSIM` parameter file specifies the fracture orientation distribution type. There are two valid choices: 1) the normal distribution, and 2) the Fisher distribution. For both options, line 36 specifies the average fracture pole trend and plunge. If the normal distribution is chosen, line 37 must specify the standard deviation of the pole trend and plunge. If the Fisher distribution is chosen, the dispersion constant, κ, must be specified on line 37.

Lines 38 to 43 specify distributions for fracture length, height and aperture. Normal, log-normal and parametric distributions may be specified on lines 38, 40 and 42. Lines 39, 41 and 43 specify the mean, standard deviation and minimum allowed value (i.e. the distribution can be truncated on the low end).

Line 44 specifies whether or not the fractures are assigned to a random z-location. If 0 (no) is specified, all fractures within a grid cell are assigned to the middle z-coordinate of that cell. This allows for fractures that fully and exactly penetrate horizontally lying strata, by setting disabling the "random z-location" and making the fracture height equal to the bed height.

5.8.2 Other Important `DFNSIM` Parameters

Line 45 of the parameter file specifies the FMF. This is the parameter that defines how many extra fractures are generated. For example, if the target fracture intensity is 1000 fractures and the FMF is 3, then 3000 fractures will be generated for the pool and 1000 are randomly assigned to the initial DFN to begin.

Lines 48 to 53 specify several parameters related to the DFN optimization. Line 48 defines whether or not to optimize the DFN at all. In some instances, the user may want to simulate a DFN with the Poisson process approach (Baecher model). The number of iteration loops on line 49 is another important parameter. This number specifies how many times each fracture is visited during the optimization to have its activation changed. The choice of the number of iteration loops was discussed previously in Section 5.6.

The `DFNSIM` user is able to optimize the DFN on any combination of the five objective function components, as specified on line 50. For example, often fracture length data is unavailable or unreliable. In this case, the user could choose not to optimize the DFN based on fracture length. Line 52 defines scaling constants, which were discussed in Section 5.3 and are used to stress the relative importance of certain spatial statistics for the DFN to honour. For example, experience has shown that it is easy to optimize fracture intensity (since all DFNs start out with the target intensity) and harder to optimize spacing. One could increase the scaling constant for spacing and/or reduce the constant for intersections.

5.9 Remarks

An algorithm for generating DFNs was proposed, which shows promise in that it allows for simulation of DFNs that match target input statistics on an alternative measure of fracture spacing, relative orientation of fractures, length, intensity and the number of intersections. The algorithm works by simulating more fractures than are required and iterating to find a subset that best matches the target input histograms. An objective function is minimized to find the best quality fit between the target statistics and those from the DFN.

The algorithm is flexible and can be used in two or three dimensions. Experience in applying the algorithm has shown that tens of millions of fractures can be simulated and optimized in a reasonable computation time (less than a day on a Dell Studio XPS 9100 with a Intel Core i7 CPU).

One shortcoming of this approach is that it requires the user to define the target histograms and the number of fracture intersections. This may not be possible in cases where limited information on the fractures is available. However, if fracture information is available from core or borehole images in at least a few wells, or two dimensional aerial images of the fractures are available, the target histograms can be calculated.

Validation of reservoir and fracture models is important. Reservoir models are often validated by a technique known as cross-validation, where a sample is left out and a value for that location is calculated using nearby data. This is repeated for each data set. Then the correlation between the actual data and the predicted values at those locations can be calculated. Higher correlations indicate better models. This would not work for DFN models since the goal is not to predict actual fracture locations. The goal here, is to get the spatial distribution of fractures and their orientations correct on a larger scale. An additional challenge is presented by the often short correlation ranges of fracture properties (Makel, 2007), which makes comparing fracture models to newly drilled wells difficult. The challenge of model validation is not considered a shortcoming of the proposed DFN simulation algorithm since it also applies to the traditional DFN simulation techniques. In part, validation of fracture models (and reservoir models in general) is accomplished through history matching and comparison of actual and predicted performance. This is somewhat unsatisfying since history matching is non-unique.

Chapter 6

Case Study: Tensleep Formation at Teapot Dome

The purpose of this chapter is to present a case study that demonstrates how the proposed DFN simulation algorithm can be used to create geologically realistic fracture networks within the context of the geomodelling work flow for a naturally fractured reservoir. The publicly available Teapot Dome data set was used for this case study.

Log, core and FMI data were examined in detail. The procedure for establishing relationships between core porosity and permeability are presented herein. FMI data was used to establish distributions of fracture properties such as intensity, aperture, size, local orientation and spacing. Available well picks were used to map horizons. Variograms were modelled and sequential Gaussian simulation was used to simulate rock matrix properties. Small-scale DFNs were simulated to model fractures at the well-bore scale. The DFNs were shown to match target histograms of local fracture spacing and orientation as well as honouring the measured fracture intensity and number of intersections. The DFNs were upscaled to an equivalent permeability tensor and fracture porosity. The upscaled equivalent fracture permeability and porosity were used as data to inform area-wide sequential Gaussian simulation of fracture permeability and porosity.

6.1 Background on Teapot Dome

Teapot Dome is located in Natrona County, Wyoming, about 30 miles north of Casper (See Figure 6.1). Full-scale development of the field began in 1976 and production

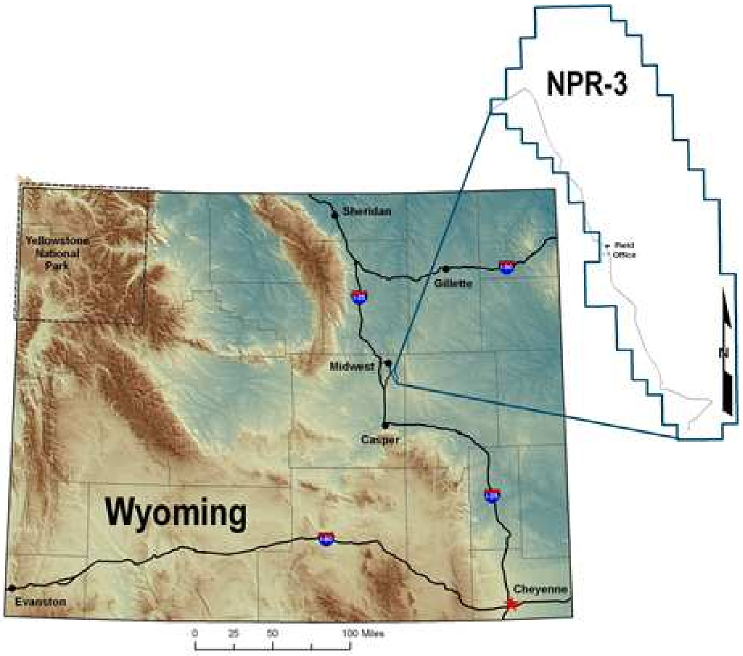
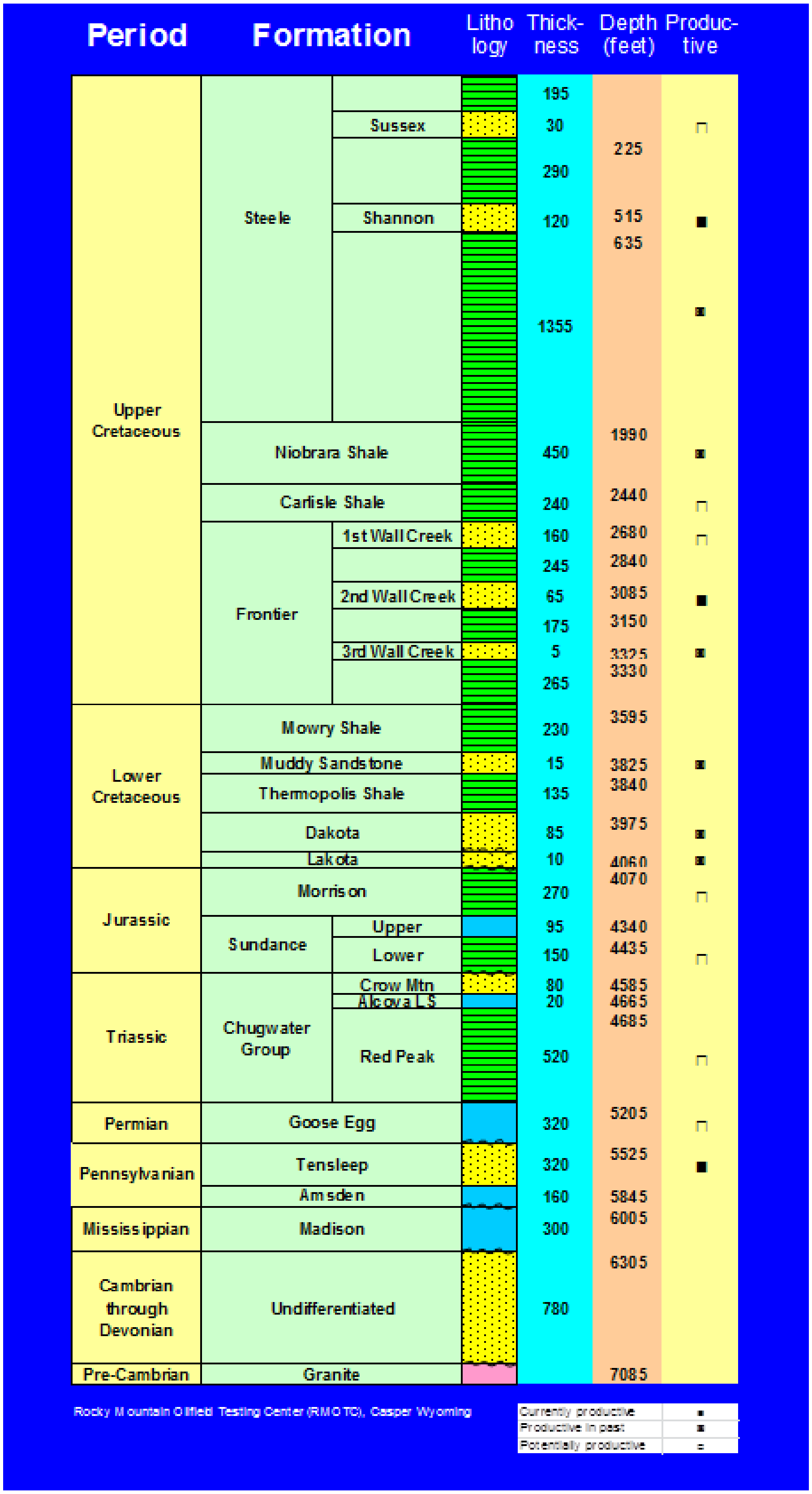


Figure 6.1: Location of the Teapot Dome Oil Field, formerly Naval Petroleum Reserve #3 (NPR-3) (RMOTC, 2009).

peaked in 1981. In 1992, the US Department of Energy created the Rocky Mountain Oilfield Testing Center (RMOTC) to manage the Teapot Dome field and converted it into a working laboratory. The RMOTC has collected a large amount of reservoir data and made it available to the public as non-proprietary data. The data was provided in imperial units and has not been changed for this research for ease of comparison with other published works (Schwartz, 2006; Smith, 2008). The results are intended to be illustrative.

Hydrocarbons were produced from nine zones at Teapot Dome. The geologic column provided by the RMOTC is shown in Figure 6.2. The Tensleep Formation consists of four units, which are (in order of increasing depth): 1) the A Sandstone, 2) the B Dolomite, 3) the B Sandstone and 4) the C1 Dolomite. Production in the Tensleep is from the A and B Sandstone units. The Tensleep sandstone has a porosity of approximately 7% and is comprised of eolian dunes.

Three types of data were used in this study: core, geophysical logs and FMIs. The area of interest shown is 9000 feet wide by 12500 feet long (2.7 by 3.8 km) and is suitable since it includes three of five wells with FMI data but is a reasonable size for modelling (See Figure 6.3). There are 29 wells with geophysical data and horizon picks located in the study area.



Period	Formation		Litho logy	Thick-ness	Depth (feet)	Produc-tive
Upper Cretaceous	Steele		🟩	195		
		Sussex	🟨	30		□
			🟩	290	225	
		Shannon	🟨	120	515	■
			🟩	1355	635	▣
	Niobrara Shale		🟩	450	1990	▣
	Carlisle Shale		🟩	240	2440	□
	Frontier	1st Wall Creek	🟨	160	2680	□
			🟩	245	2840	
		2nd Wall Creek	🟨	65	3085	■
			🟩	175	3150	
		3rd Wall Creek	🟨	5	3325	▣
			🟩	265	3330	
Lower Cretaceous	Mowry Shale		🟩	230	3595	
	Muddy Sandstone		🟨	15	3825	▣
	Thermopolis Shale		🟩	135	3840	
	Dakota		🟨	85	3975	▣
	Lakota		🟨	10	4060	▣
Jurassic	Morrison		🟩	270	4070	□
	Sundance	Upper	🟦	95	4340	
		Lower	🟩	150	4435	□
Triassic	Chugwater Group	Crow Mtn	🟨	80	4585	
		Alcova LS	🟦	20	4665	
		Red Peak	🟩	520	4685	□
Permian	Goose Egg		🟦	320	5205	□
Pennsylvanian	Tensleep		🟨	320	5525	■
	Amsden		🟦	160	5845	
Mississippian	Madison		🟦	300	6005	
Cambrian through Devonian	Undifferentiated		🟨	780	6305	
Pre-Cambrian	Granite		🟪		7085	

Rocky Mountain Oilfield Testing Center (RMOTC), Casper Wyoming

Currently productive	■
Productive in past	▣
Potentially productive	□

Figure 6.2: Teapot Dome geological column (RMOTC, 2009).

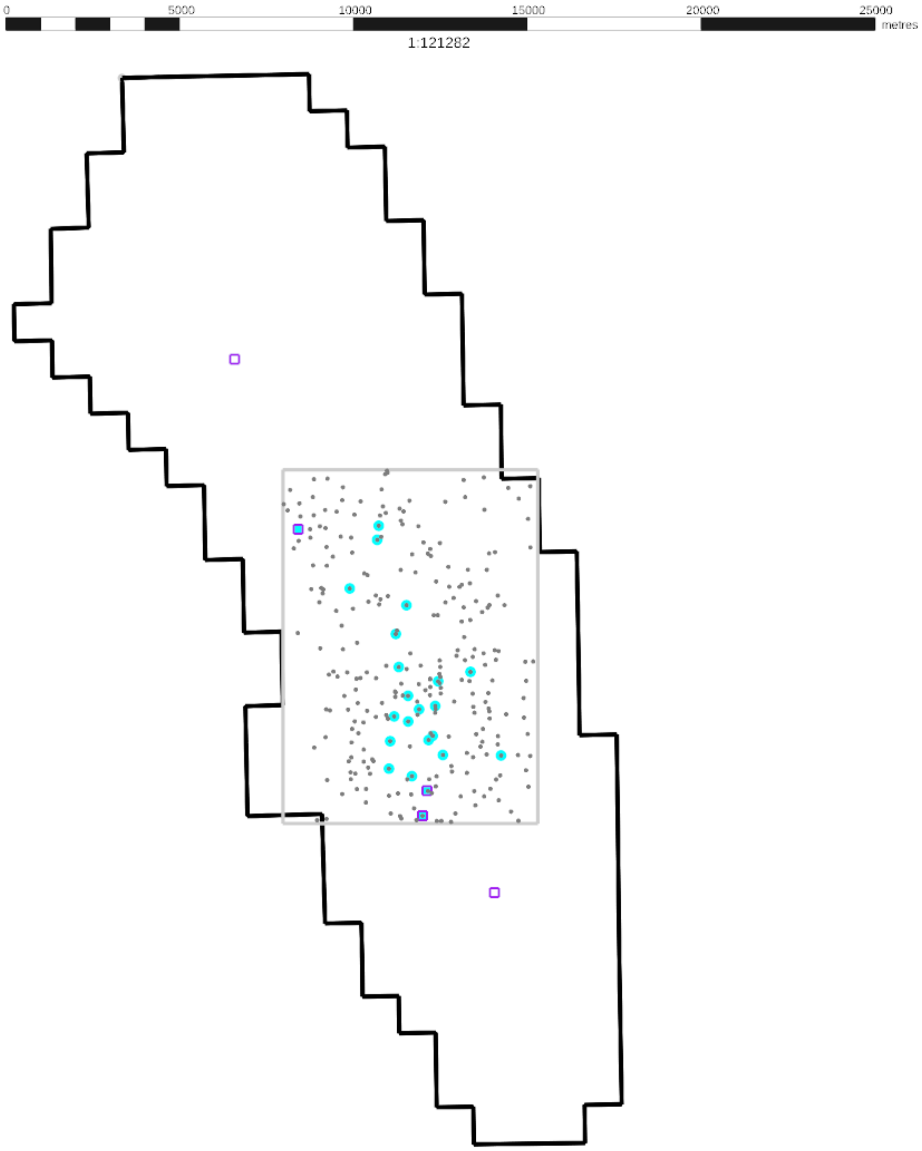


Figure 6.3: Modelling area of interest within the bounds of the Teapot Dome Oil Field. The small dots within the modelling area indicate all wells. The dots surrounded by a larger cyan outline are wells with formation picks. The wells indicated by purple squares have fracture data.

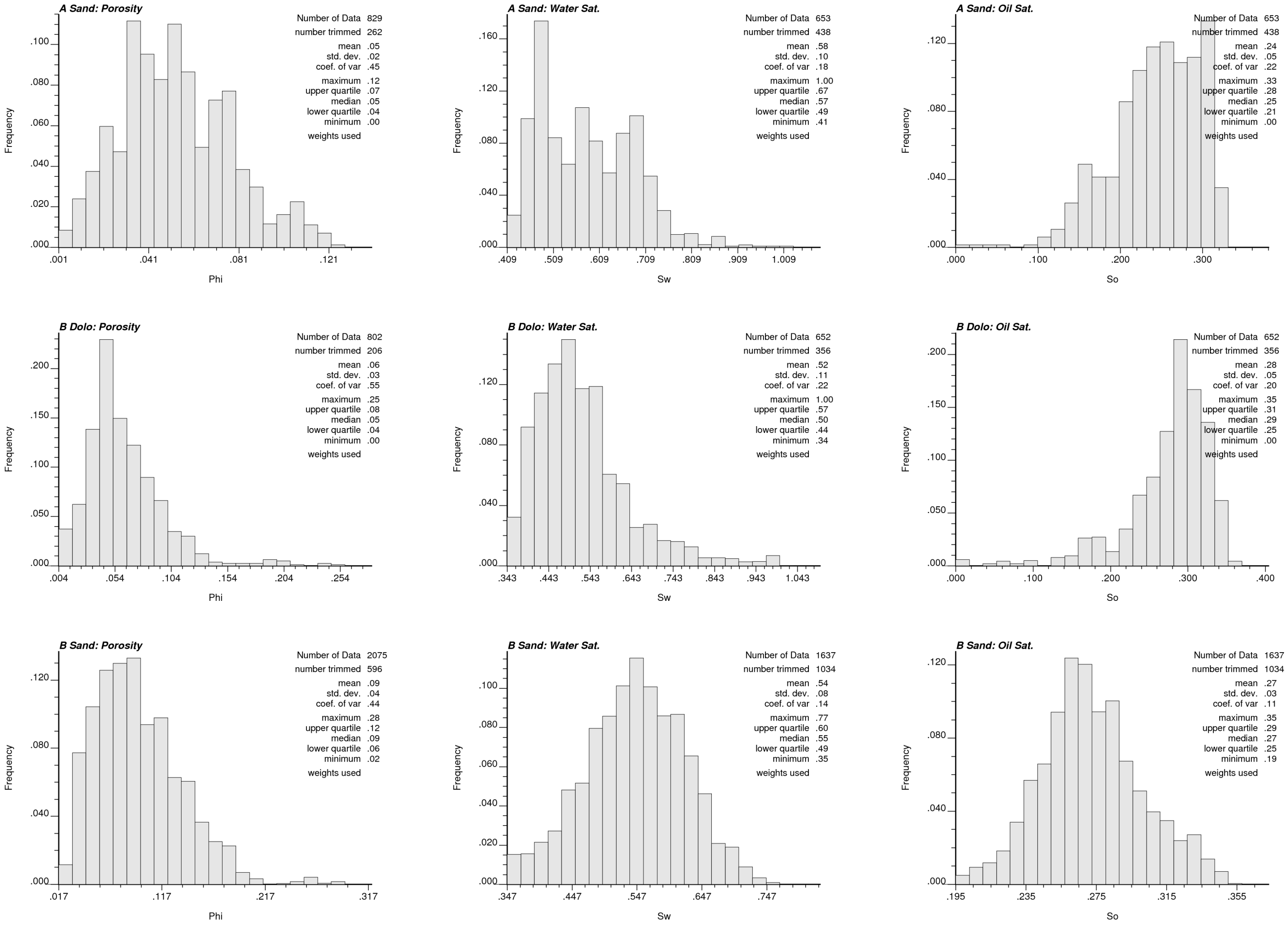


Figure 6.4: Histograms of porosity, water saturation and oil saturation for the A Sandstone, B Dolomite and B Sandstone units. Porosity, water saturation an oil saturation are measured as fractions by volume.

6.1.1 Exploratory Data Analysis

The geophysical log data was processed and extracted within each stratigraphic unit. Figure 6.4 shows histograms for porosity, water saturation and oil saturation in each of the three stratigraphic units. The average porosity increases from 5% in the A Sandstone to 6% in the B Dolomite to 9% in the B Sandstone.

Permeability sample data was only available from one well bore. Porosity data, on the other hand, is available throughout the full well bores since it is calculated from geophysical logs. Thus, if a relationship between porosity and permeability can be established, then the more densely sampled porosity data can be used to make better predictions of permeability. In this case, the available core data was used to calculate the bivariate relationships between porosity and horizontal permeability (k_h) and between porosity and vertical permeability (k_v) (see Figure 6.5) using the GSLIB program, `BIMODEL` (Deutsch and Dose, 2005). The conditional cumulative distribution functions (CCDFs) are colour coded from low (blue) to high (red) using the standard GSLIB colour scale between the limits of 0 and 1. Values less than 0.01 or greater than 0.99 are not shown. The conditional mean values are connected by a solid black line, the 0.25 and 0.75 quantiles are shown by dashed black lines and the 0.05 and 0.95 quantiles are shown by the shorter dashed lines. The conditional mean line would normally fall in the middle, but the permeability CCDFs are skewed.

These relationships between porosity and permeability were used later, along with simulations of porosity, to generate rock matrix models of horizontal and vertical permeability using a cloud transform (Deutsch and Dose, 2005).

As noted earlier, FMI logs were available for five wells. The logs were interpreted by Randy Koepsell from Schlumberger. Koepsell compiled and interpreted the FMI data to provide estimates of fracture orientation and effective hydraulic fracture aperture in addition to classifying fractures as open, partially healed and induced (Koepsell, 2001, 2002a,b,c, 2004).

Figure 6.6 shows a stereonet with all available fracture poles from within the Tensleep Formation. The contoured stereonet is also shown in the figure. The mean

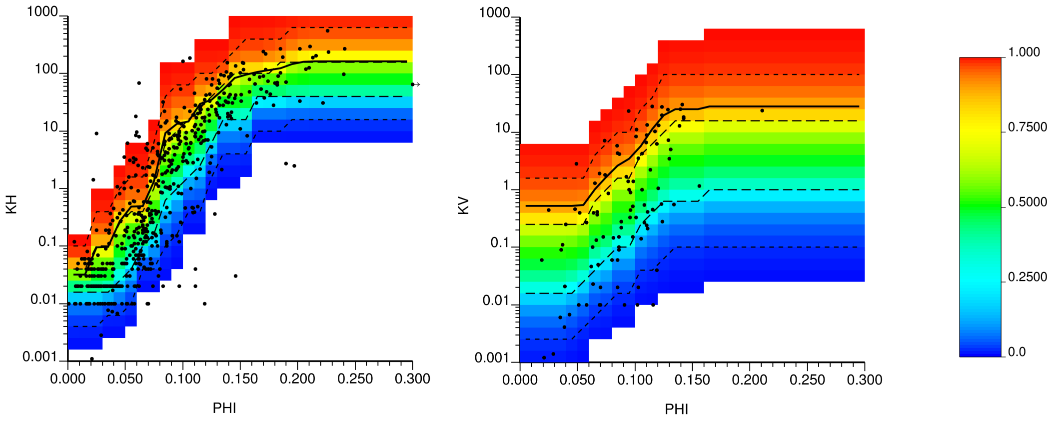


Figure 6.5: Calculated bivariate relationships between porosity and horizontal and vertical permeability. Porosity is measured as a fraction (volume/volume), KH and KV are in units of mD. Colours are indicative of the conditional cumulative distribution where red equals 1 and blue equals zero, as is indicated by the legend.

pole trend and plunge are 7 degrees clockwise from north and 10 degrees from the horizontal plane, respectively. Based on visual inspection of the contoured stereonet, the fractures were modelled as a single joint set. Studies by Cooper Cooper (2000); Cooper et al. (2006) show three joint sets oriented parallel, perpendicular and oblique to the anticline fold hinge. However, they mainly focus on the Mesaverde Formation, which is near surface whereas the Tensleep Formation is over 5500 ft below surface. The data collected from the Tensleep formation and presented in 6.6 mainly spans the range of parallel to the fold hinge with some measurements that would be classified as oblique to the fold hinge. However, there is such a concentration of poles in the parallel direction (North 10 degrees East) that, here, the fractures are modelled as one set and the variation in hinge parallel and oblique fractures are accounted for by the dispersion from the mean pole vector. The hinge-perpendicular fractures from the Mesaverde Formation are not sampled at all in the Tensleep wells.

Effective hydraulic aperture data was also available from the FMI logs. Figure 6.7 shows the average fracture aperture by geological unit. Apertures are the lowest in the tight B Dolomite and are highest in the Lower B Sandstone. Figure 6.8 shows relative frequency histograms of the natural logarithm of aperture. Also plotted on the charts are the normal distribution calculated using the mean and standard deviation of the Ln(aperture) data. The fit between the histograms and the normal distributions

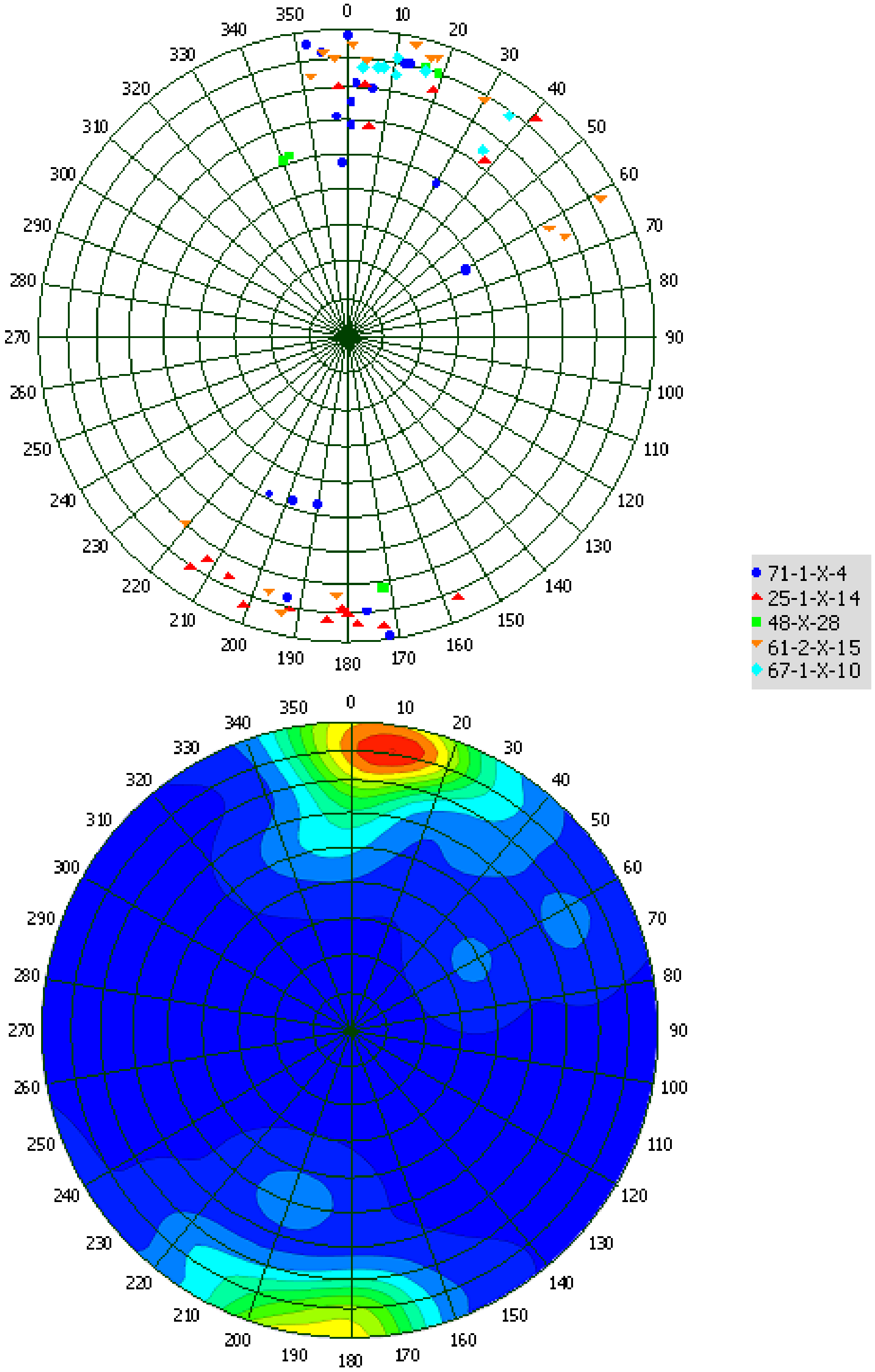


Figure 6.6: Stereonet of fracture pole orientation and contoured pole orientation.

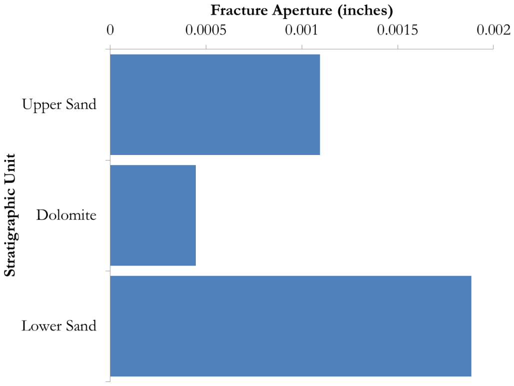


Figure 6.7: Average fracture aperture by unit.

of Ln(aperture) appears reasonable, based on visual inspection, indicating that the aperture data is distributed approximately log-normally.

Fracture intensity is defined by the dimension of the measurement region and the dimension of the fracture (Dershowitz and Herda, 1992). Common measures of fracture intensity were discussed in Section 1.2.2.

The P_{10} fracture intensity was provided in the original data set; however, the P_{32} fracture intensity is most commonly used to simulate three-dimensional DFNs (Schlumberger, 2007). The FMI data was loaded into `FracMan` (Golder Associates Ltd., 2010) (a commercial software package for analysis and modelling of fractured rock masses) in order to calculate the P_{10} fracture intensity. `FracMan` also calculates an equivalent P_{32} fracture intensity from the P_{10} intensity through Monte Carlo simulation. This functionality is based on the work by Wang (2005) on stereological relationships between fracture orientation and fracture intensity. Table 6.1 shows the number of fractures per unit and borehole as well as the P_{10} and P_{32} fracture intensities. As is shown in the table, the middle B Dolomite unit shows the highest fracture intensity. This makes sense since dolomite is more brittle than sandstone.

The Teapot Dome dataset includes data on trace length measured from the FMIs. It is possible to estimate the fracture size distribution from the trace length distribution (Ozkaya, 2003). This is made possible when not all fractures are imaged all the way around the borehole. This means that some fracture traces are partial sinusoids rather than full ones. The trace length ratio is defined as the ratio between the length of the

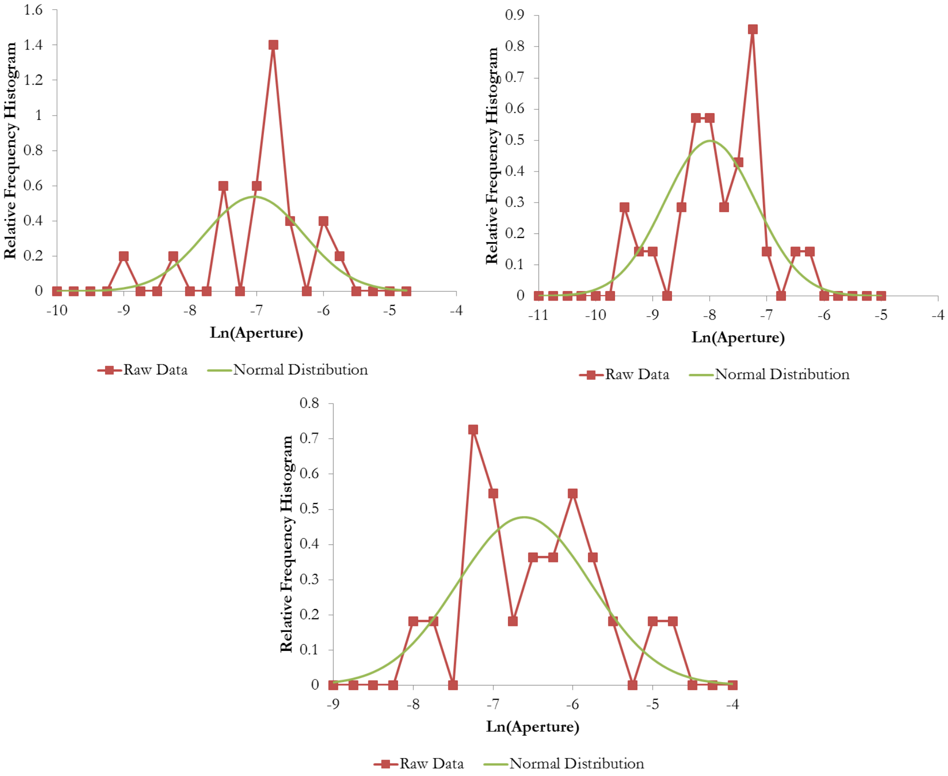


Figure 6.8: Relative frequency histograms of Ln(aperture) (red line with squares) compared with normal distributions using the mean and standard deviation of Ln(aperture) in units of Ln(inches).

Table 6.1: Fracture intensity data for Tensleep Formation fractures.

Well:	**25-1-X-14**	**48-X-28**	**61-2-X-15**	**67-1-X-10**	**71-1-X-4**	**Average**
# of Fractures						
Upper Sand	5	5	2	0	7	4
Dolomite	12	8	1	7	0	6
Lower Sand	4	5	3	9	2	5
Total	21	18	6	16	9	14
$\mathbf{P_{10}}$ (ft^{-1})						
Upper Sand	0.178	0.185	0.141	0.000	0.258	0.152
Dolomite	0.628	0.400	0.048	0.355	0.000	0.286
Lower Sand	0.067	0.082	0.049	0.150	0.039	0.078
Average	0.291	0.222	0.080	0.168	0.099	0.172
$\mathbf{P_{32}}$ (ft^{-1})						
Upper Sand	1.994	4.418	0.147	0.000	5.183	2.348
Dolomite	8.564	31.063	0.296	10.369	0.000	10.058
Lower Sand	4.859	3.551	0.192	1.955	0.742	2.260
Average	5.139	13.011	0.212	4.108	1.975	4.889

partial fracture trace and the full sinusoid. The trace length ratio is a function of the fracture orientation distribution, well geometry (borehole diameter and orientation) and the fracture size distribution. The distribution of fracture length can be estimated through simulation with different size distributions. The estimate of the fracture size distribution is non-unique, but often narrowly bounded (Golder Associates Ltd., 2010).

`FracMan` (Golder Associates Ltd., 2010) automates the calculation of the fracture size distribution from trace length data. There are some assumptions that must be made in order to estimate the size distribution:

- All discontinuities are planar. This assumption is common in practice, as is evidenced by the commercial fracture simulation software which all simulate planar fractures. At any rate, fracture curvature is often negligible (Warburton, 1980).
- Fractures are elliptical in shape and all fractures have the same aspect ratio, k.
- The fracture centroids are distributed randomly and independently in space (i.e.

by a Poisson distribution).

- The fracture size is independent of their spatial location.

Zhang et al. (2002) provide a lengthy discussion on each of these assumptions and they are deemed reasonable in most cases. `FracMan` calculates the mean and standard deviation of the trace length distribution and converts them to the mean (μ_a) and standard deviation ($(\sigma_a)^2$) of fracture radius distribution depending on an assumption of distribution shape (see Table 6.2). In the table, the parameter, M is calculated as:

$$M = \frac{\sqrt{\tan^2 \beta + 1}}{\sqrt{k^2 \tan^2 \beta + 1}}, \tag{6.1}$$

where k is the aspect ratio of the fracture (the ratio of the major to minor axes) and β is the angle between the discontinuity major axis and the line parallel to the trace.

For the Tensleep trace length data, the log-normal distribution is a better fit than the power law distribution or any of the other distributions. The mean and standard deviation of the log-normal fracture radius distribution are 19.4 and 12.9 inches, respectively. It was assumed that the Tensleep fractures would have the same mean and standard deviation. Making this assumption in order to calculate fracture size is considered reasonable given that other fracture attributes are much more important to fracture permeability than lateral fracture size (particularly aperture since flow is related to the cube of aperture). In addition, there was no other data available (i.e. outcrops) and it was thought that some estimate of fracture size, despite the uncertainties, is better than none.

In addition to using `DFNSIM` to create DFNs, it can also be used to build the target histograms of local fracture spacing and orientation. The fractures that are indicated on the FMIs can be input into `DFNSIM` and the program will build the histograms of local fracture spacing and orientation (see Section 5.5 for more information on how these distributions are calculated). These two histograms are calculated for each of the five wells and are used as target distributions when simulating the DFNs in a later step.

Table 6.2: Expressions for determining μ_a and σ_a from μ_l and σ_l after Zhang et al. (2002)

Distribution of $g(a)$	μ_a	$(\sigma_a)^2$
Log-normal	$\frac{128\mu_l^3}{3\pi^3 M\left(\mu_l^2+\sigma_l^2\right)}$	$\frac{1536\pi^2\left(\mu_l^2+\sigma_l^2\right)\mu_l^4-128^2\mu_l^6}{9\pi^6 M^2\left(\mu_l^2+\sigma_l^2\right)^2}$
Negative exponential	$\frac{2\mu_l}{\pi M}$	$\left[\frac{2\mu_l}{\pi M}\right]^2$
Gamma	$\frac{64\mu_l^2-3\pi^2\left(\mu_l^2+\sigma_l^2\right)}{8\pi M\mu_l}$	$\frac{\left[64\mu_l^2-3\pi^2\left(\mu_l^2+\sigma_l^2\right)\right]\left[3\pi^2\left(\mu_l^2+\sigma_l^2\right)-32\mu_l^2\right]}{64\pi^2 M^2\mu_l^2}$

Evaluating the number of fracture intersections between fractures measured in directly in core might be possible. Fracture extents (length and height) could be simulated from the previously estimated log-normal size distribution. The simulated fracture extents can be applied to measured fractures in place of real fracture size data. One could make the assumption that the fracture centroids lie along the borehole. That is, the x,y fracture centroid locations are at the same coordinates as the x,y location of the borehole. Or, one could randomize the fracture centroid locations such that the simulated fractures still honour the measured fractures from core (i.e. move the location of the simulated fracture along the plane of the measured fracture). One set of simulated fracture extents results in one realization of a DFN that honours the well data and has a certain number of intersections, which can easily be calculated. Multiple realizations can be generated by re-simulating the extents of each measured fracture. In this way, the average number of fracture intersections could be calculated *for the fractures measured in core.*

Applying the simulated number of fracture intersections to some volume of the model seems difficult. The core samples a certain volume rock, which measures a certain number of fractures. While it may be possible to simulate a number of fracture intersections for the fractures measured in the core volume, those intersections will mostly occur outside the volume of the core within some larger, but unknown, volume of rock. Thus, extrapolating the number of intersections inferred by a core volume to some larger volume seems problematic. If the volume of core was increased by some amount, new fractures might be measured, which might intersect some of the fractures measured by the original core volume.

6.1.2 Horizon Mapping

Horizon picks were available in 29 wells in the area of interest. Directional experimental variograms were calculated from the surface elevation data. Figure 6.9 shows the experimental variograms and their corresponding models. The red dots correspond to a direction of 160 degrees, clockwise from North. The black dots correspond to a direction of 70 degrees clockwise from North. The models are Gaussian structures with zero nugget effect and ranges of 4500 and 3000 feet in the 160 degree and 70 degree directions, respectively. The models provide a reasonable fit to the experimental variograms.

The horizons were mapped using the well picks as data. Simple global kriging (Neufeld and Wilde, 2005) was used to create maps of the horizon elevations. The advantage of simple global kriging is that it does not produce any artefacts related to the search since no search is used. All data values are used in the calculation of each estimate. The calculated horizons are shown in Figure 6.10.

Between each of the four mapped horizons, isopachs are calculated (see Figure 6.11). The average thickness of the A Sandstone, B Dolomite and B Sandstone units are 25 feet, 22 feet and 60 feet, respectively.

6.2 Stratigraphic Coordinates

Reservoir units are defined to provide a large-scale subdivision of the reservoir into geologically homogeneous units. Each unit is defined by a top and base surface grid. Differential compaction, structural deformation, erosion or subsequent deposition filling existing topography results in deformed geologic units and surfaces (Deutsch, 2002). As a result, the continuity of the facies and reservoir properties may not follow a grid model that corresponds to existing surfaces.

One solution is to model the facies and reservoir properties in a transformed coordinate space with stratigraphic coordinates. A full discussion of reservoir gridding

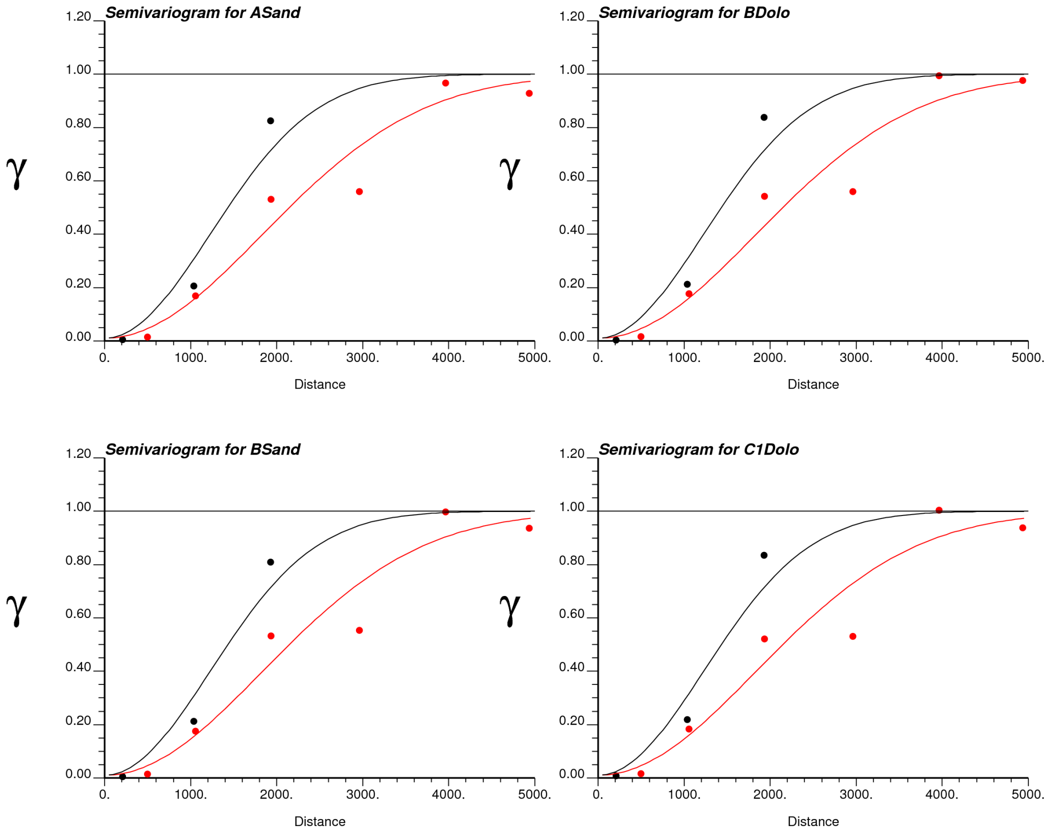


Figure 6.9: Modelled directional experimental variograms for each of the four horizons. The experimental variograms are dotted and the models are the solid lines.

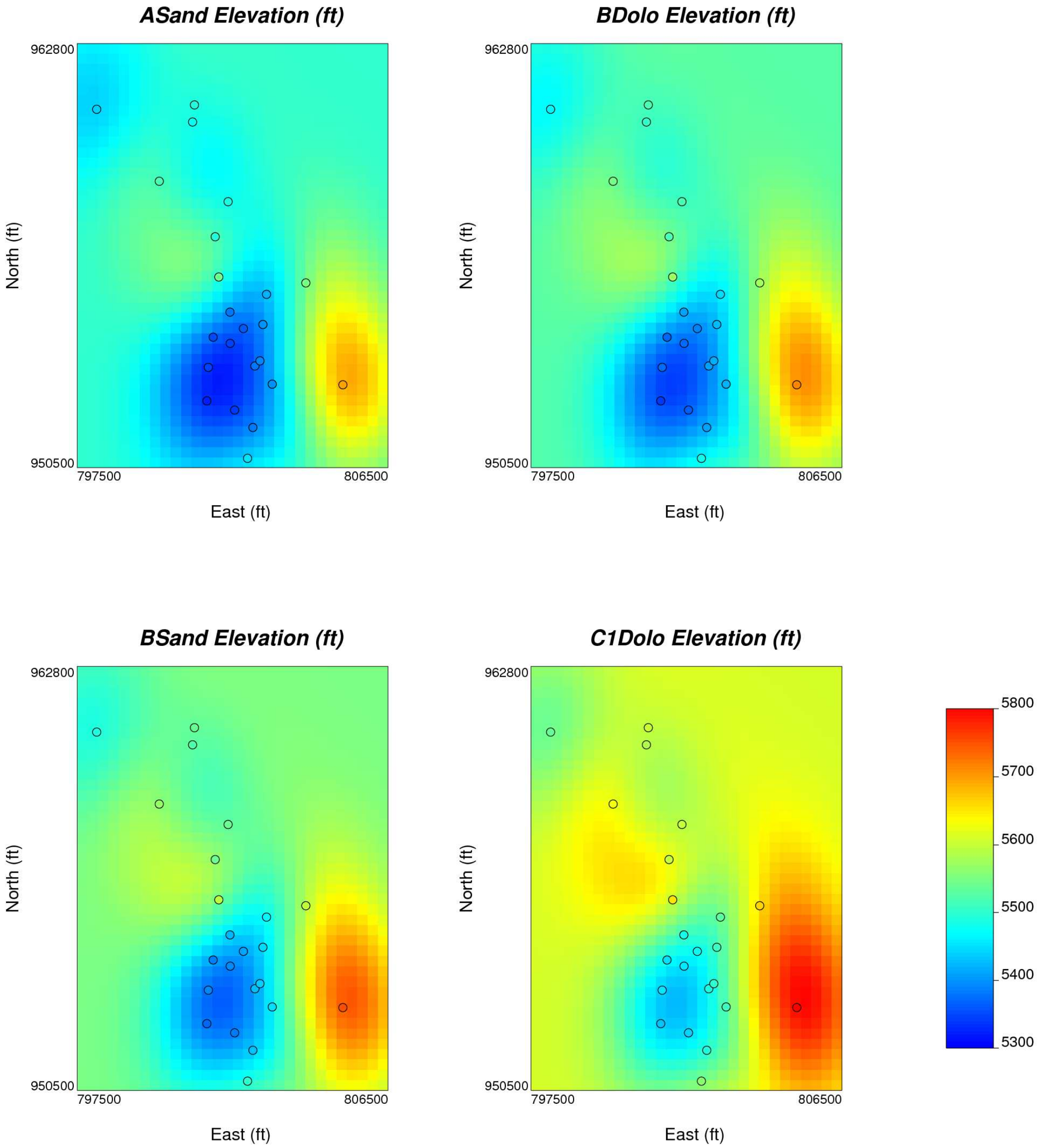


Figure 6.10: The four kriged stratigraphic surfaces (elevation in feet above sea level). Well locations with stratigraphic markers are indicated by the circles.

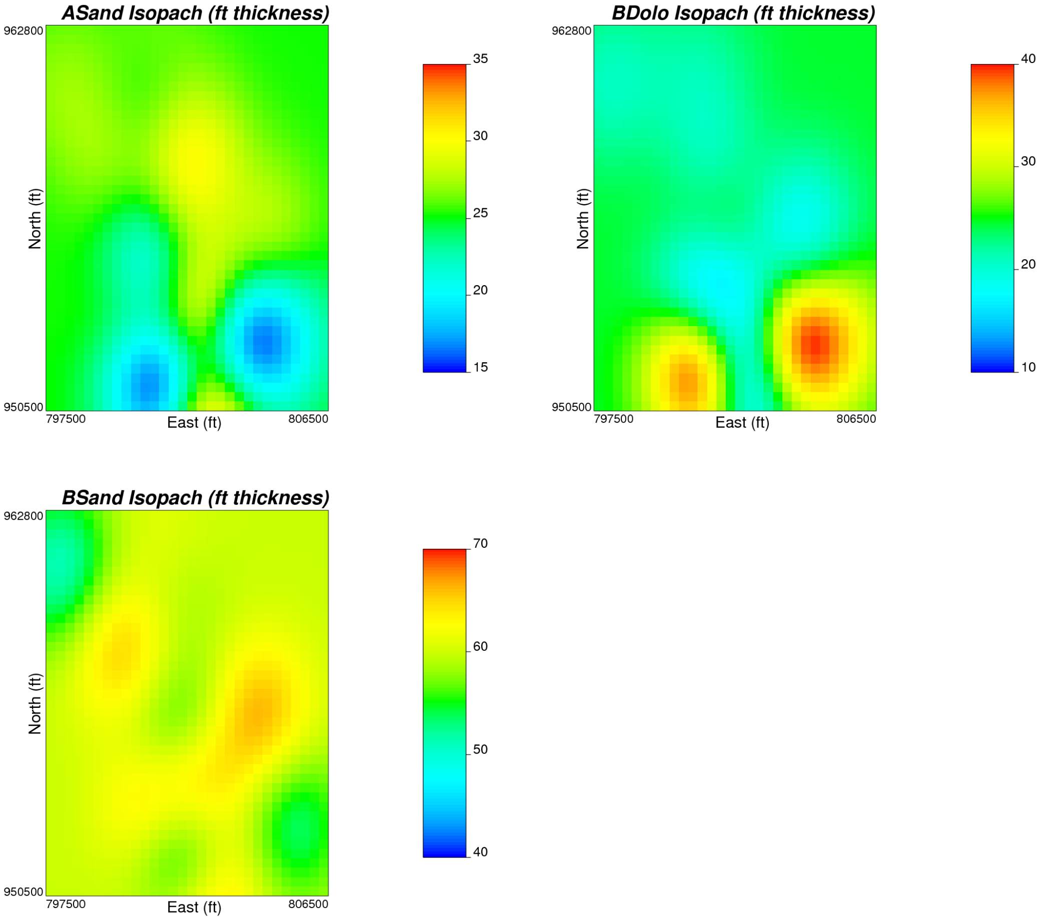


Figure 6.11: The three isopachs of the geological unit thickness (in feet)

and stratigraphic coordinate calculation is outside the scope of this example; interested readers are referred to Deutsch (2002). In this case, there was no available geological data indicating whether the surfaces are subject to erosion or filling phenomenon. Since this case study is mainly illustrative in nature, it was assumed that the strata conform to the existing top and base surfaces. Thus, vertical proportional coordinates were calculated to replace depth or elevation data. Vertical proportional coordinates are the relative distance between a correlation top surface and base and are calculated as follows:

$$z_{rel} = \frac{z - z_{cb}}{z_{ct} - z_{cb}} T \tag{6.2}$$

where:

- z_{rel} is zero at the stratigraphic base surface and T at the stratigraphic top;
- T is the thickness constant equal to the average thickness of the unit;
- z_{cb} is the correlation base and z_{ct} is the correlation top.

Converting all depth measurements to proportional coordinates allows modelling of each reservoir layer in regular Cartesian coordinates. For this study, a vertical proportional coordinate was calculated for each geophysical log data within each of the three reservoir units.

6.3 Rock Matrix Modelling

All subsequent analysis and rock matrix modelling took place in the depositional space defined by the proportional stratigraphic coordinates.

Many reservoirs are not evenly sampled. Usually boreholes are drilled preferentially in areas with good reservoir quality. As a result, the well log data often not representative of the entire reservoir. In these cases the data must first be "declustered". That is, weights are calculated for each data and a weighted histogram is calculated for the variable of interest. A full discussion on declustering is outside the scope of this thesis, but Deutsch (2002) provides plenty of detail on the subject for interested readers. For the well data collected in the study area, cell declustering of the data was investigated and found to make no significant difference. That is, the weighted (declustered) mean values were nearly identical to the unweighted (raw) mean values for each variable. Cell declustering was used since it is popular, simple and reliable (Deutsch, 2002).

Next, the data was transformed to Gaussian space, which is required for all Gaussian based geostatistical techniques such as sequential Gaussian simulation, spectral simulation and turning bands.

Variograms were calculated on the Gaussian porosity, water saturation and oil saturation data within each of the three reservoir units. Figure 6.12 shows the experimental variograms (red dots) and their fitted models (black lines) for the A Sandstone. The variograms for the B Dolomite and B Sandstone are shown in Appendix A. Spherical models were used since they appeared to provide the best fit for the experimental variograms. No nugget effect was modelled since the vertical variograms show no nugget. Directional variograms were calculated to detect any obvious anisotropy, but none was found.

The rock properties were simulated using sequential Gaussian simulation (SGS) (Deutsch, 1992). SGS is one of the most popular simulation methods for continuous variables. It is used in a number of popular geomodelling software packages, such as Petrel. Spectral simulation (used in Roxar's RMS geomodelling software) or turning bands (used in the Isatis geostatistics software) could also have been used with little practical difference to the results.

In order to avoid data from one geological unit unfairly influencing simulated values in another unit, the geological units are simulated separately. The rock properties were simulated on a grid that has 30 x 50 x 25 cells in the x, y and z directions, respectively. The grid cells were 300 feet in the x and y directions and 1 foot in the z (vertical) direction. The grid specifications are chosen as a balance between computation time and resolution. More cells may be better but take longer to simulate.

After simulation, the modelled variables were back-transformed to the original unit space. Figure 6.13 shows two cross sections through each the simulated models for A Sandstone porosity, water saturation and oil saturation. The realizations of porosity, water saturation and oil saturation for the B Dolomite and B Sandstone are shown in Appendix A in Figures A.3 and A.4.

Very limited data is available on the matrix rock permeability. In fact, the Teapot Dome data set includes permeability data from only one well. However, porosity data (derived from gamma ray geophysical measurements) are available in 29 wells. If a relationship between porosity and permeability can be established where data for both properties exist, then porosity can be used as a predictor for permeability in wells where no permeability data is available.

One could model a fitted relationship (such as a best fit line) between permeability and porosity. Then for every unique value of porosity, there would be a unique value of permeability. However, the correlation between porosity and permeability is far from perfect (see Figure 6.5) and this technique ignores this uncertainty. The cloud transform (Deutsch and Dose, 2005) was used instead. The cloud transform randomly draws from the conditional distribution of permeability given a simulated porosity value. The conditional distributions of matrix permeability given porosity are shown in Figure 6.5. Figure 6.14 shows cross sections through a realization of horizontal and vertical permeability in the A Sandstone. The realizations of permeability are less continuous than the porosity realizations due to the cloud transform technique. The realizations of permeability for the B Dolomite and B Sandstone are shown in Appendix A in Figures A.5 and A.6.

6.4 Fracture Modelling with Non-Random DFNs

The area of interest is 9,000 feet by 12,500 feet (2.7 km by 3.8 km). The average P_{32} fracture intensity in the 22 ft thick middle B Dolomite unit is 10.1 $\mathrm{m}^2/\mathrm{m}^3$. The grid volume is 2.5x10^9 ft^3 (6.8x10^7 m^3). If the average fracture is a square with the dimensions 20 by 20 feet, this means that the P_{32} intensity of one fracture is 1.6x10^{-7} m^2/m^3. If the target intensity is 10.1 m^2/m^3, this means that 62 million fractures are needed to achieve the desired target intensity over the entire area of interest. It would take a number of days to calculate and optimize a DFN of this size. Thus, instead of simulating a full-field DFN, well-bore scale DFNs were calculated at the location

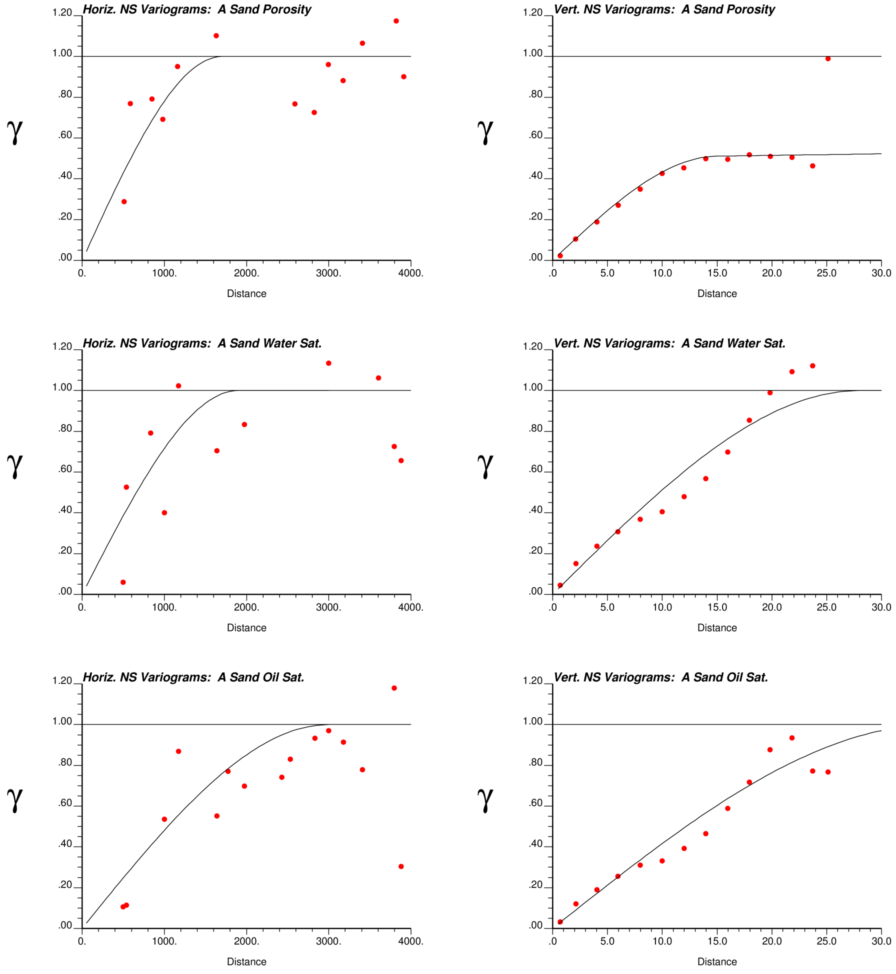


Figure 6.12: Horizontal and vertical normal score semi-variograms for the A Sandstone Unit (top row = porosity, middle row = water saturation, bottom row = oil saturation). Distance is measured in feet.

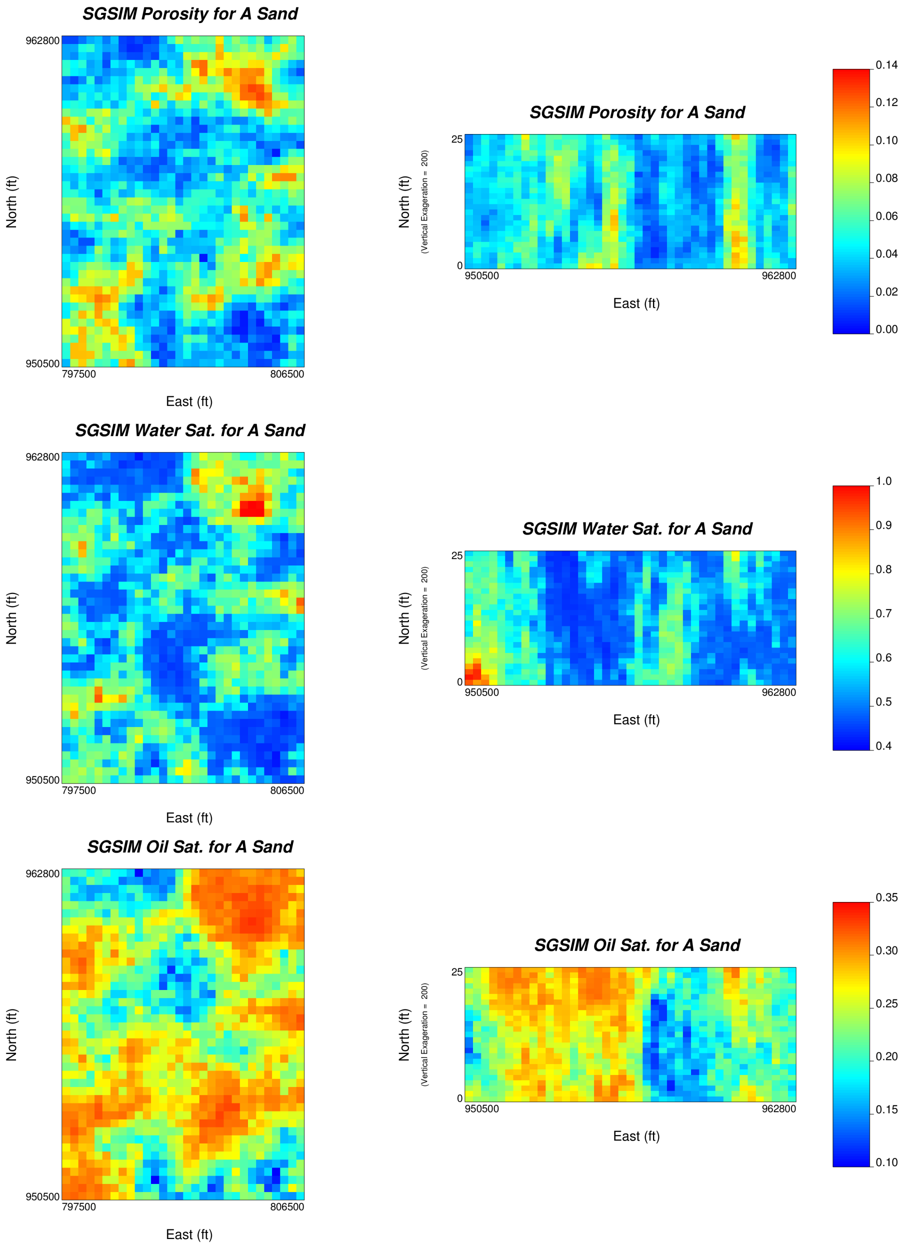


Figure 6.13: Cross sections of one realization of porosity, water saturation and oil saturation through the A Sandstone Unit.

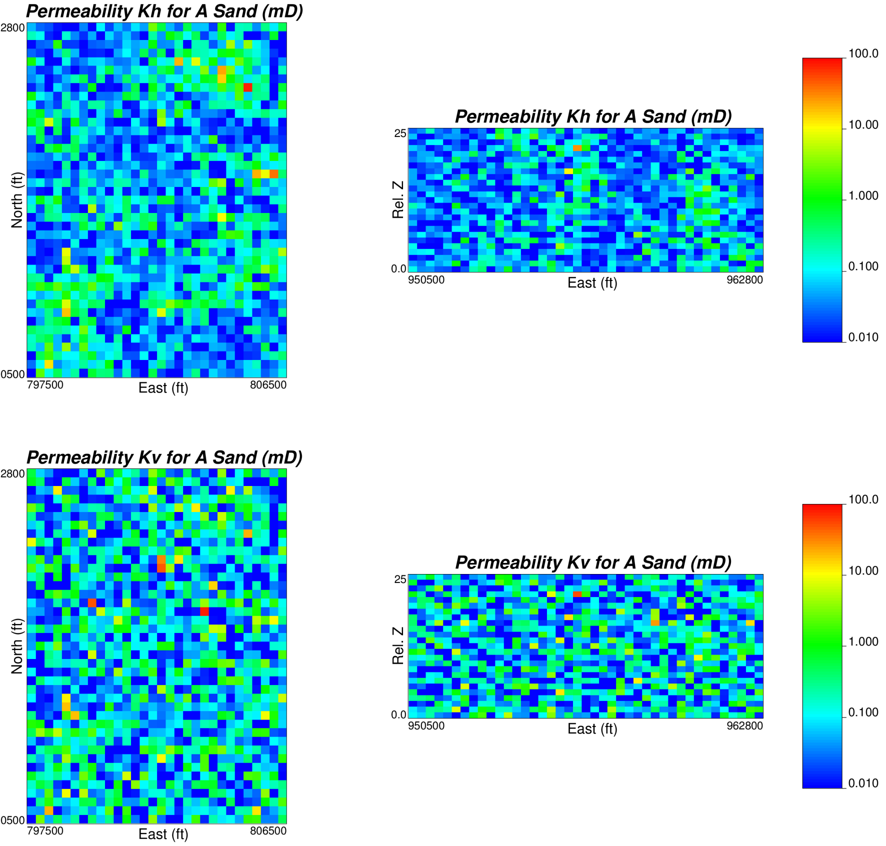


Figure 6.14: Cross sections of one realization of horizontal and vertical permeability in the A Sandstone Unit.

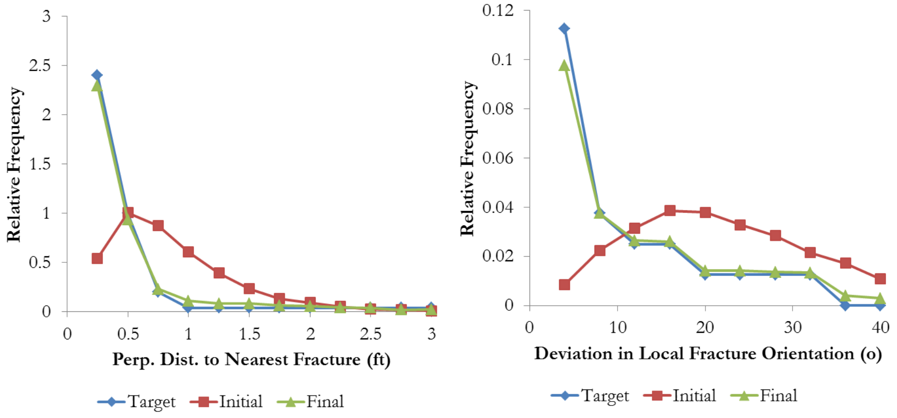


Figure 6.15: Target, Initial DFN and Final DFN histograms of local fracture spacing (left) and deviation in local fracture orientation (right) for well 25-1-X-14 in the A Sandstone unit.

of the five wells with FMI data. Note, the P_{32} was used for these calculations (and the DFN modelling) because it is the most common measure of intensity for modelling fractures in three dimensions (Schlumberger, 2007). This is discussed in more detail in the introductory chapter.

`DFNSIM` was used to calculate the optimized well bore scale DFNs. The DFNs were simulated in a region that is one geocellular grid cell in size in the x and y-directions (300 x 300 ft) and the thickness of the unit in the z-direction. Fractures were simulated using the input distributions of orientation, size and aperture identified by the exploratory data analysis. The DFNs were optimized so that the final DFNs closely match the target distributions of local fracture spacing and orientation and intensity. In this case the DFNs are not optimized on the number of fracture intersections since no information on that parameter is available, as is discussed earlier. Figure 6.15 shows the target, initial DFN and final DFN histograms of local fracture spacing and local fracture orientation for well 25-1-X-14 in the A Sandstone unit. The target and final histograms are near perfect matches. Moreover, the final P_{32} fracture intensity matches the target by within 2.2%. Figure 6.16 shows an example of a DFN for well 25-1-X-14 in the A Sandstone. The target, initial DFN and final DFN histograms for the other wells in the A Sandstone unit, along with those of the wells in the B Dolomite and B Sandstone units are shown in Appendix A.

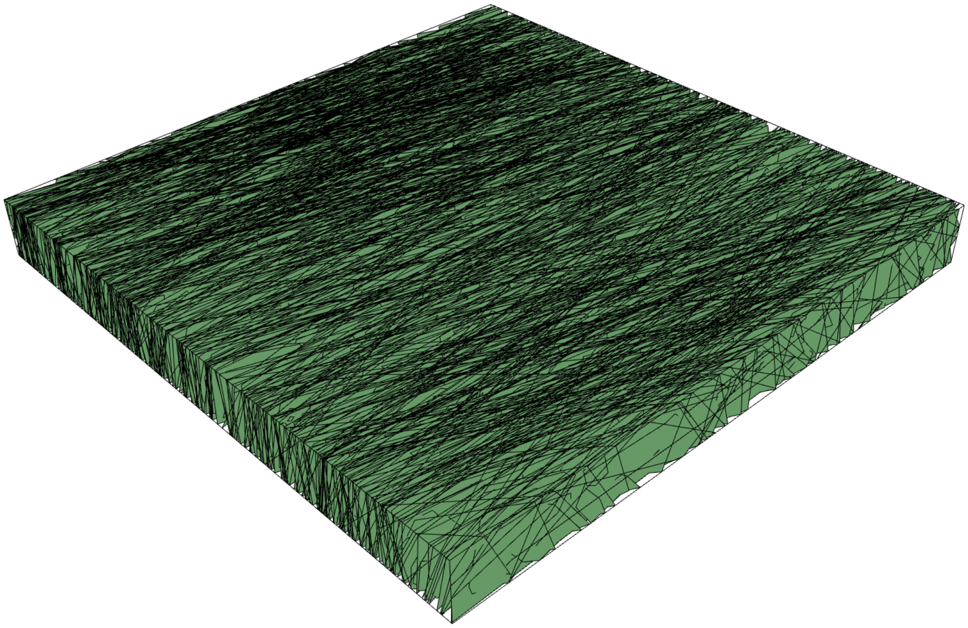


Figure 6.16: An example DFN for well 25-1-X-14 within the A Sandstone unit. The extents of the grid cell is 300 ft in the x and y-directions and 30 ft in the z-direction. The P_{32} fracture intensity is 2.0 ft^2/ft^3.

6.5 DFN Upscaling

DFN upscaling methods are discussed in some detail in Section 1.2.2. Oda's method was used for this case study because it is simple and quick to calculate. It does not consider the interconnectedness of the fracture network, but this assumption is considered acceptable for well connected fracture networks (Dershowitz et al., 2004), as in the case of Teapot Dome, which is shown visually in Figure 6.16.

Equivalent fracture permeability tensors (K_{xx}, K_{yy} and K_{zz}) and fracture porosity were calculated for the well bore scale DFNs. These well bore scale equivalent permeability tensors were used as data in the next step to inform field-wide fracture permeability and porosity simulations.

The DFNs were simulated in original coordinate space because the transform to proportional coordinate space would require an awkward transform of the fracture orientation distribution. The upscaled equivalent fracture porosity and permeability tensor was then easily transformed to proportional coordinates using the same methodology described earlier. Thus, fracture property modelling took place in the same proportional coordinate space as the rock matrix properties. This means that both models were simulated on the same geocellular grid, allowing them to feed into a flow simulator.

6.6 Field-Wide Fracture Permeability and Porosity Simulation

The equivalent fracture permeability and fracture porosity well-bore scale data were transformed to Gaussian space (as noted earlier, this is required for subsequent geostatistical simulation). Variograms were calculated in the vertical direction using the Gaussian data. The vertical variograms for K_{xx}, K_{yy} and K_{zz} and fracture porosity are shown in Figure 6.17 for the A Sandstone. Here, the vertical variograms indicate the presence of a nugget effect, which was included in the models. The variograms for the other two units are similar and are shown in Appendix A. The horizontal variograms are inaccessible since there are only five wells with fracture information. Thus, a horizontal to vertical anisotropy ratio of 100:1 was assumed. Deutsch (2002) indicates that ratios between 50:1 and 250:1 are reasonable.

Separate equivalent property models of fracture porosity and permeability were simulated for each of the three geological units. For the same reasons as in the rock matrix property modelling, SGS was also used for this modelling of the four fracture parameters. Each simulation variable used their respective modelled variograms. The Gaussian simulated data was then back transformed to original unit space. One realization of each of the four fracture variables for the A Sandstone is shown in Figure 16. Realizations of permeability and porosity in the other units are shown in Appendix A. Fracture porosity is quite small since the fractures themselves have very small apertures.

6.7 Remarks

The case study example in this chapter is mainly illustrative in nature. In theory, it would be ideal to show a comparison between reservoir scale DFNs built using the more traditional Poisson based approaches and the proposed fracture simulation algorithm. Computational requirements associated with simulating and optimizing so many frac-

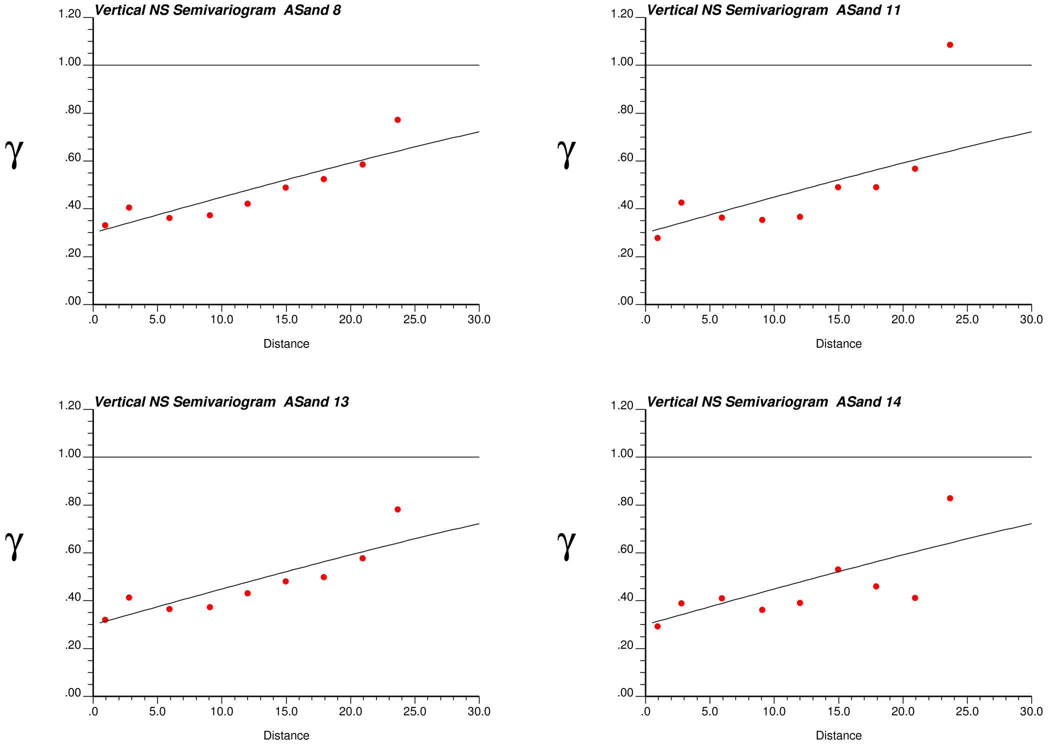


Figure 6.17: Vertical semi-variograms for K_{xx} (top left), K_{yy} (top right), K_{zz} (bottom left) and fracture porosity (bottom right) in the A Sandstone unit. The red dots are the experimental variograms. Distance is measured in feet.

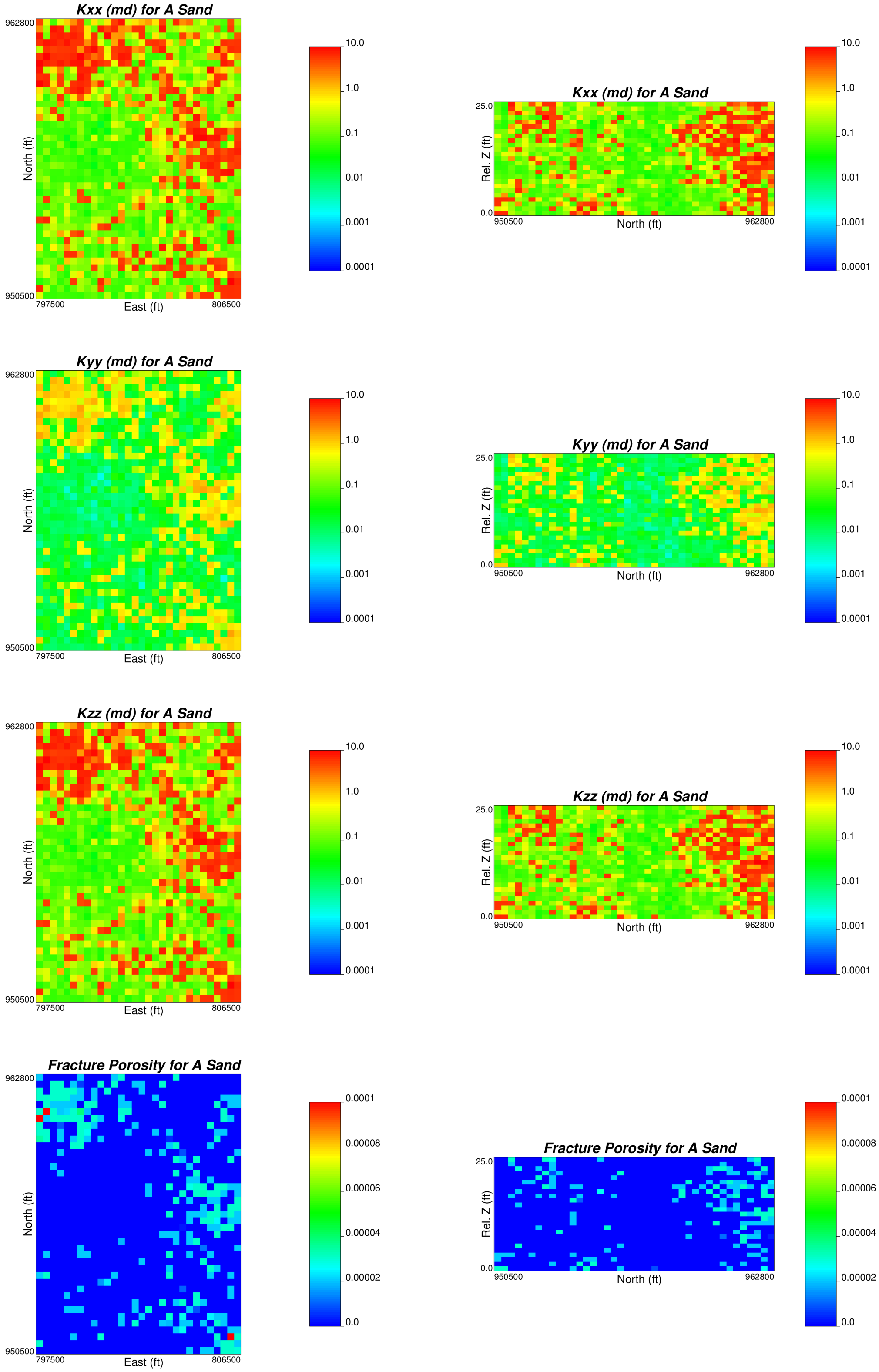


Figure 6.18: One realization of the simulated K_{xx}, K_{yy}, K_{zz} and fracture porosity in the A Sandstone unit. The realizations of fracture permeability have units of millidarcies (mD) while fracture porosity is a volume fraction.

tures make that impractical. DFNs were simulated at the scale of the well bore. The inter-well region was simulated with sequential Gaussian simulation. Unfortunately, the inter-well simulation of fracture properties removes nearly all of the influence of the target fracture network spatial statistics. The histograms of perpendicular distance to the nearest fracture and deviation in local orientation of fractures are not honoured in the inter-well locations, making a comparison to traditional DFNs unsatisfying. Therefore, this case study is included in order to demonstrate the application of the proposed methodology for simulating DFNs and its associated `DFNSIM` computer code to modelling a real-world naturally fractured reservoir. This case study shows that the proposed methodology can be used to generate geologically realistic models of reservoir fractures at the well-bore scale. Along with the rock matrix property models, the simulated models of fracture porosity and permeability can be used as input into a dual-porosity flow simulator for further analysis.

Chapter 7

Concluding Remarks

This thesis addressed selected topics related to the geostatistical modelling of naturally fractured reservoirs. The goal is to obtain geologically realistic geomodels that integrate all data and quantify uncertainty. This is important because more accurate reservoir models permit improved reservoir management and development decisions, which can have significant economic consequences for oil companies.

7.1 Summary of Contributions

A novel robust correlation coefficient was proposed along with a methodology for calculating its uncertainty. It was shown that some natural fracture networks show a pattern of fracture spacing and deviation in local fracture orientation that cannot be reproduced in DFNs created using with industry standard Poisson-based DFN modelling algorithms. A new DFN simulation algorithm was proposed that generates DFNs that honour histograms of local fracture spacing, deviation in local fracture orientation, the number of fracture intersections, fracture length, and fracture intensity. The contributions of this thesis are expounded upon below.

7.1.1 On the Proposed Robust Correlation Coefficient

Relationships between bivariate data are frequently quantified using the correlation coefficient. Traditional means of calculating experimental correlation coefficients are known to be adversely affected by outlier data. A new method for calculating a robust correlation coefficient was proposed based on a weighted average correlation calculated from different combinations, or subsets, of the original data. The proposed robust correlation coefficient was shown to have a higher breakdown point than either Pearson's or Spearman's correlation coefficients as well as two other robust correlation coefficients.

When the sample size is small, the uncertainty in the measured correlation can be very large, especially when the measured correlation is low. A methodology to calculate the uncertainty in the correlation coefficient was proposed using its sampling distribution. The sampling distribution for the correlation coefficient requires the measured correlation and the number of independent data. Earth sciences data is often spatially dependent, which means that often the data are not truly independent. Thus, a methodology for calculating an effective number of independent data using the variogram was proposed.

7.1.2 On the Randomness of Natural Fracture Networks

This thesis also considered the randomness of natural fracture networks. Typical DFN simulation software considers fracture centroids as a Poisson process (either homogeneous or non-homogeneous). Two natural fracture networks were digitized and DFNs were simulated as models of those fractures. The fracture centroids in the simulated DFNs were generated using a Poisson process, as is typical in industry and the literature. This thesis compared the natural fracture networks to the simulated DFNs and showed that the simulated DFNs were substantially different from the natural fracture networks. The simulated DFNs showed substantially different histograms of fracture spacing and deviation in local fracture orientation compared to the natural fracture networks. As well, the simulated DFNs showed far more fracture intersections than the natural fracture networks. One of the goals of modelling fracture networks with DFNs is to simulate geologically realistic fracture networks and this work shows that this may not be possible using the typical Poisson process-based simulation algorithms.

7.1.3 On the Simulation of DFNs Conditional Subject to Constraints

This thesis proposes a new algorithm for DFN simulation. The goal of the proposed DFN simulation algorithm is to generate DFNs that honour certain fracture network spatial statistics such as:

- the histogram of perpendicular distance to the nearest fracture;
- the histogram of deviation in local fracture orientation;
- the histogram of fracture length;
- the number of fracture intersections in the DFN; and,
- the target fracture intensity.

The algorithm works by generating more fractures than are required and retaining a subset that minimizes the difference between the target fracture network spatial statistics and those of the DFN.

The proposed algorithm is flexible and can be used in two or three dimensions. Tens of millions of fractures can be simulated and optimized in a reasonable computation time (less than a day on modern computers).

7.2 Limitations

There are some limitations to implementing the ideas put forth in this thesis.

In calculating the proposed robust correlation coefficient, the number of subsets for which correlation coefficients must be calculated grows exponentially with the number of data. For cases where the number of data exceed approximately 20, the calculation time becomes prohibitively large. As a result, it was proposed to randomly sample a certain number of data combinations instead of calculating correlations for all possible data combinations. However, if the number of data is much larger than 20, this may result in only sampling a small fraction of the total number of data combinations. Thus, the proposed robust correlation coefficient may be more appropriate for smaller data sets for which correlations for all possible combinations of data can be calculated. On the other hand, the code for the `ROBUSTCORRCO` program automatically calculates the proposed robust correlation coefficient along with three other robust correlation coefficients. It is recommended that the user compare the results of the different methods to arrive at a reasonable value.

This thesis showed that some natural fracture networks are not well modelled by typical DFN simulation algorithms. However, the natural fracture networks considered were two dimensional only (lineaments or fracture traces). This was largely unavoidable since the three-dimensional spatial distribution of fractures in a rock mass are not known. Although the Teapot Dome dataset is three-dimensional, there are only five wells with fracture information and the true natural fracture network remains largely unknown. Regardless, this is not considered to be a major source of error. As long as the geomodeller can calculate reliable three-dimensional spatial statistics for the natural fracture network, the proposed fracture simulation algorithm is capable of honouring that information in an optimized DFN.

The use of an alternative measure of fracture spacing (the perpendicular distance to the nearest fracture) as opposed to more conventional measures of fracture spacing is considered to be a limitation of the DFN simulation algorithm. The proposed DFN simulation algorithm is able to generate DFNs that honour histograms of perpendicular distance to the nearest fracture; however, these DFNs do not necessarily honour the histograms of other fracture spacing measures.

The two-dimensional fracture networks presented in this thesis were fully known. As a result, it was easy to generate target spatial statistics for the fracture networks (i.e. histograms, the number of intersections and fracture intensity). In the context of petroleum reservoir modelling, three-dimensional fracture networks must be considered. The Teapot Dome case study presented in Chapter 7 showed that generating the target spatial statistics is generally possible, but with some difficulties. For one, it becomes difficult, if not impossible, to determine an appropriate target number of fracture intersections for a three-dimensional DFN when intersections are only measured along a one-dimensional line (the well-bore). It might be possible to determine the number of fracture intersections between fractures that are measured at the well-bore by drawing fracture lengths from their distribution and assigning them to those known fractures and measuring the number of intersections between those fractures. However, determining the number or probability of fracture intersections beyond the area very close to the well-bore may not be possible. It is possible that analogue data or some form of geomechanical model based on first principles would aid in predicting the number of fracture intersections.

This work also does not consider the type of fracture intersections between sets. For example, when a new fracture grows and reaches an old fracture it may terminate at the old fracture plane (a Y-node) or propagate through the old fracture (an X-node). Typically some proportion of fracture intersections will be X, Y and I-nodes (fractures that do not intersect any others). The main reason that fracture terminations were not considered is that it was well investigated by Dershowitz (1985) and implemented in the enhanced Baecher model for fracture simulation.

Although experience with `DFNSIM` has shown that it is possible to generate fracture networks with tens of millions of fractures in less than a day on modern computers, some field-scale NFRs will have more fractures. This may restrict the use of the proposed algorithm either to well-bore scale DFNs or only considering fractures within a certain size range. This is a limitation of other fracture simulation techniques as well. Even though computation time using other fracture simulation algorithms is lower than for the algorithm proposed in this thesis, some reservoirs have fracture sizes ranging from kilometre scale down to sub-millimetre scale. Representing the smaller fractures becomes impractical and is usually not needed. Nevertheless, it is a limitation that the computational expense of the optimization in the proposed algorithm restricts the number of fractures that may be simulated more than for simpler approaches.

7.3 Future Areas of Research

The presented approach to geostatistical modelling of NFRs could benefit from further research. In particular, the choice of using an alternative measure of fracture spacing in the proposed DFN simulation algorithm could be re-visited. The reasons in favour of an alternative measure of fracture spacing are discussed in Section 5.2 and were mainly related to computational speed. However, it may be possible to speed up the re-calculation of true fracture spacing by considering a constant set of imaginary sampling lines, rather than calculating new random scan lines each time a fracture is added or removed from the DFN. Perhaps the fracture spacings can be calculated for

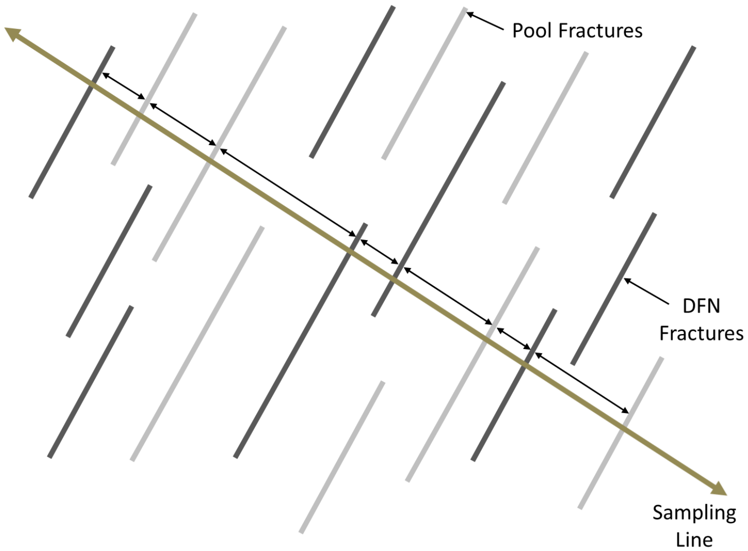


Figure 7.1: A constant scan line propagating through fractures that are part of the DFN and the remaining fractures in the pool.

the entire pool of fractures with one realization of sampling lines. This means that for any subset of fractures, the spacings between the fractures along the sampling lines would already be known as a combination of the smaller spacings between the pool fractures. Figure 7.1 illustrates this concept. Each time a fracture is activated from the pool, the true spacings can be locally updated since their component distances are already known and stored. The resulting DFNs would honour the true fracture spacing histograms, rather than needing to rely on an alternative measure of fracture spacing.

The relative number of the different types of fracture intersections (I, X and Y-nodes) was not considered (see Figure 7.2). This issue could be further explored and implemented within the framework of the proposed algorithm. Instead of considering the overall number of intersections, the objective function could consider the number of each type of fracture intersection separately. Or perhaps, the objective function could consider the "distance" on a ternary diagram between the target proportions of fracture intersection types and the actual proportion of fracture intersection types.

7.4 Final Remarks

Oil companies want to make optimal decisions regarding reservoir development, production and management. Failure to do so can have significant economic consequences.

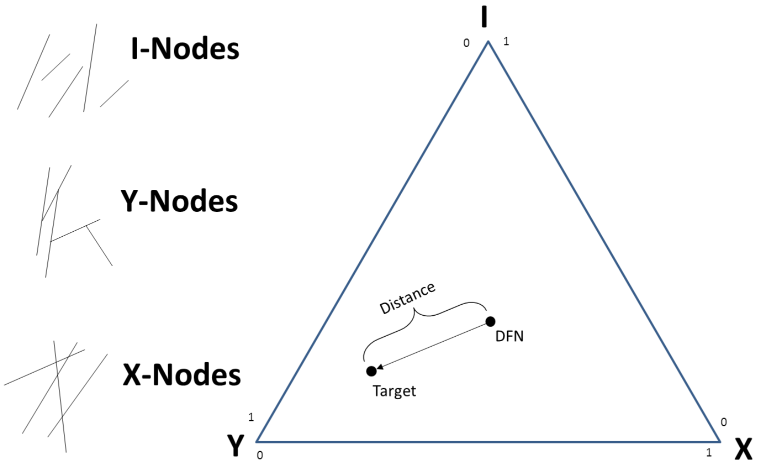


Figure 7.2: Fracture intersection types. Nodal proportions are captured in a ternary diagram (modified from Makel (2007)).

The list of ways in which poor decisions cause economic loss is long. A few examples include: drilling too many wells or not enough wells or drilling wells in the wrong locations, making poor predictions of reservoir performance, not correctly predicting water breakthrough, etc.

The basis for making good decisions in a petroleum reservoir context starts with a fundamental understanding of geology. Even if geological knowledge and understanding is high, there is still significant uncertainty in reservoir rock types and properties in inter-well locations. After all, only a tiny amount of the reservoir rock is actually sampled by core. Predictions at inter-well locations are made using geostatistical techniques. Those predictions and geomodels must be 1) as accurate as possible, incorporating all available information, and 2) they must capture the uncertainty in those predictions caused by incomplete knowledge of the geology and rock properties. Accurate geomodels that capture uncertainty help support correct reservoir decision making.

This thesis makes contributions to the field of geostatistical modelling of NFRs that result in more geologically realistic and accurate geomodels with a better representation of the true uncertainty.

Secondary data (such as seismic) is often incorporated in geomodels using methods that rely heavily on the correlation coefficient. Using a robust correlation coefficient that is less sensitive to the effects of outlier data results in more accurate geomodels. Furthermore, considering the uncertainty in the correlation coefficient and propagating that through geostatistical modelling helps geologists and engineers better understand the uncertainty in their reservoirs.

Traditional means of simulating DFNs account for fracture size, intensity and orientation data but do not explicitly account for fracture spacing data, the relative orientation of nearby fractures and the number of intersections. Thus, DFNs created to model NFRs may show a greater degree of connectedness than is supported by data. This thesis proposed a new methodology for DFN simulation that honours additional spatial statistics that are not captured by traditional methods. This has major implications for flow simulation and may help account for discrepancies between NFR flow predictions and actual performance. When reservoir data supports the interpretation of non-random distributions of fracture locations and orientations, it is essential to account for those distributions in order to arrive at the most accurate DFNs and reservoir models possible. More accurate reservoir models lead to better reservoir decisions and increased profits.

It should be noted that the techniques and methodologies proposed in this thesis are not only be applicable to NFRs. The robust correlation coefficient and its uncertainty can be applied to nearly any situation where Pearson's correlation's coefficient is applied. Considering the uncertainty in correlation coefficients will also be useful in many cases using any sort of data sets. The proposed fracture simulation algorithm could also be useful in other situations where realistic fracture models are required. Fractures are a major concern in both underground and surface mining. If the data show certain spacing and local orientation distributions, any models not honouring those spatial statistics will show unrealistic intact block sizes. In the case where fracture spacing is approximately log-normal (fractures have a tendency to not form extremely close together) and they are oriented similarly to their nearest neighbours, current fracture simulation techniques may generate smaller block sizes than are realistic. This could result in either 1) unsafe mining conditions caused by recommendations stemming from geotechnical models that do not honour all available data, or 2) costly remediation of slopes or underground passageways that may not have been needed.

Bibliography

Abdullah, M. B. (1990). On a robust correlation coefficient. *The Statistician*, 39:455–460.

Angerer, E., Lanfranchi, P., Pharez, S., and S., R. (2004). Integrated approach for seismic fracture characterisation for improved reservoir performance. In *11th Abu Dhabi International Petroleum Exhibition and Conference, SPE 88677*.

Baecher, G. (1983). Statistical analysis of rock mass fracturing. *Mathematical Geology*, 15(2):329–348.

Baecher, G., Lanney, N., and Einstein, H. H. (1977). Statistical description of rock properties and sampling. In *Proceedings of the 18th U.S. Symposium on Rock Mechanics*. American Institute of Mining Engineers.

Baker, R. O. and Kuppe, F. (2000). Reservoir characterization for naturally fractured reservoirs. In *SPE Annual Technical Conference and Exhibition, SPE 63286*.

Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data*. Wiley, 3rd edition.

Basquet, R., Jeannin, L., Lange, A., Bourbiaux, B. J., and Sarda, S. (2004). Gas-Flow simulation in discrete Fracture-Network models. *SPE Reservoir Evaluation & Engineering, SPE 88985*, 7(5):378–384.

Becker, A. and Gross, M. (1996). Mechanism for joint saturation in mechanically layered rocks: an example from southern israel. *Tectonophysics*, 257(2-4):223–237.

Belfield, W. C. (1998). Incorporating spatial distribution into stochastic modelling of fractures: multifractals and levy-stable statistics. *Journal of Structural Geology*, 20(4):473–486.

Beydoun, Z. R. (1998). Arabian plate oil and gas: why so rich and so prolific? *Episodes*, 21(2):74–81.

Bourbiaux, B. J., Basquet, R., Cacas, M. C., Daniel, J., and Sarda, S. (2002). An integrated workflow to account for Multi-Scale fractures in reservoir simulation models: Implementation and benefits. In *10th Abu Dhabi International Petroleum Exhibition and Conference, SPE 78489*.

Bourbiaux, B. J., Basquet, R., Daniel, J. M., Hu, L. Y., Jenni, S., Lange, A., and Rasolofosaon, P. (2005). Fractured reservoirs modelling: a review of the challenges and some recent solutions. *First Break*, 23:33–40.

Bourne, S. J., Rijkels, L., Stephenson, B. J., and Willemse, E. J. M. (2000). Predictive modelling of naturally fractured reservoirs using geomechanics and flow simulation. In *Abu Dhabi Interntional Petroelum Exhibition and Conference, SPE 87253*.

Bridges, M. (1975). Presentation of fracture data for rock mechanics. In *Proceedings of the Second Australia-New Zealand Conference on Geomechanics*, pages 144–148. Australian Institute of Engineers.

Cacas, M. C., Daniel, J. M., and Letouzey, J. (2001). Nested geological modelling of naturally fractured reservoirs. *Petroleum Geoscience*, 7(S):S43–S52.

Campbell, N. A. (1980). Robust procedures in multivariate analysis i: Robust covariance estimation. *Applied Statistics*, 29(3):231–237.

Casciano, C., Ruvo, L., Volpi, B., and Masserano, F. (2004). Well test simulation through discrete fracture network modelling in a fractured carbonate reservoir. *Petroleum Geoscience*, 10(4):331–342.

Chiles, J. P. and Delfiner, P. (2012). *Geostatistics: Modelling Spatial Uncertainty*. John Wiley and Sons, Inc.

Clemo, T. (1994). *Dual Permeability Modeling of Fractured Media*. Ph.D. thesis, University of British Columbia.

Cooper, S. (2000). Deformation within a basement-cored anticline: Teapot Dome, Wyoming. M.Sc. thesis, New Mexico Tech, Socorrow, N.M.

Cooper, S., Goodwin, L., and Lorenz, J. (2006). Fracture and fault patterns associated with basement-cored anticlines; the example of teapot dome, wyoming. *AAPG Bulletin*, 90(12):1903–1920.

Davies, L. and Gather, U. (1993). The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792.

Davis, J. C. (2002). *Statistics and Data Analysis in Geology*. Wiley and Sons, New York, 3rd edition.

Dershowitz, W. S. (1985). *Rock joint systems*. Ph.D. thesis, Massachusetts Institute of Technology.

Dershowitz, W. S. (1988). Characterizing rock joint geometry with joint system models. *Rock Mechanics and Rock Engineering*, 21:21–51.

Dershowitz, W. S. and Herda, H. H. (1992). Interpretation of fracture spacing and intensity. In *33rd US Symposium on Rock Mechanics*, pages 757–766.

Dershowitz, W. S., Pointe, P. R. L., and Doe, T. W. (2004). Advances in discrete fracture network modeling. In *Proceedings of the US EPA/NGWA Fractured Rock Conference*, pages 882–894.

Deutsch, C. V. (1992). *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*. PhD thesis, Stanford University.

Deutsch, C. V. (2002). *Geostatistical Reservoir Modeling*. Oxford University Press, Inc., New York, New York.

Deutsch, C. V. and Dose, T. (2005). Programs for debiasing and cloud transformation: `BIMODEL` and `CLTRANS`. In *Seventh Annual Report of the Centre for Computational Geostatistics*, University of Alberta.

Deutsch, C. V. and Journel, A. (1998). *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, second edition edition.

Doyen, P. (1988). Porosity from seismic data: A geostatistical approach. *Geophysics*, 53(10):1263–1275.

Edwards, R. V. (2006). *Processing Random Data: Statistics for Engineers and Scientists*. World Scientific Publishing Co., Singapore.

Eikmans, J., McKean, H., and Hooper, R. (1999). Using 3D integrated modelling to manage the fracture Natih field (Oman). In *SPE Middle East Oil Show, SPE 53227*, Bahrain.

Emerson Process Management (Roxar) (2011). Roxar RMS. Software.

Gauthier, B. D. M., Garcia, M., and Daniel, J. M. (2002). Integrated fractured reservoir characterization: A case study in a North Africa field. *SPE Reservoir Evaluation & Engineering, SPE 79105*, 5(4):284–294.

Ghosh, A. and Daemen, J. (1993). Fractal characteristics of rock discontinuities. *Engineering Geology*, 34:1–9.

Gideon, R. A. and Hollister, R. A. (1987). A rank correlation coefficient resistant to outliers. *Journal of the American Statistical Association*, 82(398):656–666.

Gilman, J. (2003). Practical aspects of simulation of fractured reservoirs. In *International Forum on Reservoir Simulation*.

Golder Associates Ltd. (2010). FracMan7: interactive discrete feature, data analysis, geometric modeling and exploration simulation. Software.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation*. Oxford University Press.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21.

Guaiquirian, L., Gonzalez, P., Gonzalez, A., Hernandez, M. M., Maux, T. L., Mattioni, L., and Rouvroy, P. (2007). Use of discrete fracture network ”DFN” to characterise and model a naturally fractured sandstone reservoir: A case study of Orocual field, San Juan formation, Venezuala. In *2007 SPE Latin American and Caribbean Petroleum Engineering Conference, SPE 108052*.

Hakami, E. and Barton, N. (1990). Aperture measurements and flow experiments using transparent replicas of rock joint systems. In *Proceedings of the international symposium on rock joints*, pages 383–390, Rotterdam.

Hartley, L. J. (1998). NAPSAC (Release 4.1) technical summary document. Technical report, AEA Technology.

Hawkins, D. M. (1980). *Identification of Outliers*. Chapman and Hall, London.

Heffer, K., King, P., and Jones, A. (1999). Fracture modelling as part of integrated reservoir characterization. Bahrain. Society of Petroleum Engineers, SPE 53347.

Hosseinian, A., Rasouli, V., and Bahrami, H. (2010). Analytical and numerical analysis of fluid flow through rough natural fracture profiles. In *44th US Rock Mechanics Symposium*, Salt Lake City, Utah.

Huang, Q. and Angelier, J. (1989). Fracture spacing and its relation to bed thickness. *Geological Magazine*, 126:355–362.

Isaaks, E. H. and Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics*. Oxford University Press, Inc., New York.

Iwano, M. and Einstein, H. H. (1993). Stochastic analysis of surface roughness, aperture and flow in a single fracture. In *Proceeings of the international symposium EUROCK '93*, pages 135–141, Lisbon.

Ji, S. and Saruwatari, K. (1998). A revised model for the relationship between joint spacing and layer thickness. *Journal of Structural Geology*, 20(11):1495–1508.

Johns, R., Steude, J., Castanier, L., and Roberts, P. (1993). Nondestructive measurements of fracture aperture in crystalline rock cores using x-ray computed tomography. *Journal of Geophys. Res.*, 98(B2):1889–1900.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995). *Continuous Univariate Distributions*, volume 2. John Wiley and Sons, Inc., New York, second edition.

Johnson, R. A. and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*. Pearson prentice Hall, Upper Saddle River, NJ, 6th edition.

Journel, A. G. and Zhang, T. (2006). The necessity of a multiple-point prior model. *Mathematical Geology*, 38(5):591–610.

Kalkomey, C. T. (1997). Potential risks when using seismic attributes as predictors of reservoir properties. *The Leading Edge*, pages 247–251.

Kazemi, H. and Gilman, J. (1993). *Flow and Contaminant Transport in Fractured Rock*, chapter Chapter 6: Multiphase Flow in Fractured Petroleum Reservoirs, pages 267–323. Academic Press, San Diego, CA.

Kazemi, H., Merril, L., Porterfield, K., and Zeman, P. (1976). Numerical simulation of water-oil flow in naturally fractured reservoirs. *Society of Petroleum Engineering Journal*, pages 317–326.

Kim, J. and Fessler, J. A. (2004). Intensity-based image registration using robust correlation coefficients. *Medical Imaging, IEEE Transactions on*, 23(11):1430–1444. ID: 1.

Kim, T. H. and Schechter, D. S. (2007). Estimation of fracture porosity of naturally fractured reservoirs with no matrix porosity using fractal discrete fracture networks. In *SPE Annual Technical Conference and Exhibition, SPE 110720*.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Koepsell, R. (2001). Fracture analysis of fmi log for well 61-2-x-15 for the department of energy. Technical report, Schlumberger Oil Field Services.

Koepsell, R. (2002a). Fracture analysis of fmi log for well 25-1-x-14 for the department of energy. Technical report, Schlumberger Oil Field Services.

Koepsell, R. (2002b). Fracture analysis of fmi log for well 67-1-x-10 for the department of energy. Technical report, Schlumberger Oil Field Services.

Koepsell, R. (2002c). Fracture analysis of fmi log for well 71-1-x-4 for the department of energy. Technical report, Schlumberger Oil Field Services.

Koepsell, R. (2004). Fracture analysis of fmi log for well 48-x-28 for the department of energy. Technical report, Schlumberger Oil Field Services.

Koudina, N., Garcia, R. G., and Thovert, J. F. (1998). Permeability of Three-Dimensional fracture networks. *Physical Review E*, 57(4):4466–4479.

La Pointe, P. (1988). A method to characterize fracture density and connectivity through fractal geometry. *International Journal of Rock Mechanics and Mining Science*, Abstr. 25:421–429.

Ladeira, F. and Price, N. (1981). Relationship between fracture spacing and bed thickness. *Journal of Structural Geology*, 3(2):179–183.

Lantuejoul, C. (2002). *Geostatistical simulation: models and algorithms*. Springer.

Luthi, S. M. and Souhaite, P. (1990). Fracture apertures from electrical borehole scans. *Geophysics*, 55:821–833.

Lyster, S. (2009). *Simulation of Geologic Phenomena Using Multiple-Point Statistics in a Gibbs Sampler Algorithm*. Ph.D. thesis, University of Alberta, Edmonton, Alberta, Canada.

Makel, G. H. (2007). The modelling of fractured reservoirs: constraints and potential for fracture network geometry and hydraulics analysis. *Geological Society, London, Special Publications*, 292(1):375–403.

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature*. W.H. Freeman and Company, New York.

Mitrani, I. (1982). *Simulation techniques for discrete event systems*. Cambridge Computer Science Texts: 14. Cambridge University Press, Cambridge.

Narr, W., Schechter, D. S., and Thompson, L. B. (2006). *Naturally Fractured Reservoir Characterization*. Society of Petroleum Engineers, Richardson, Texas.

Narr, W. and Suppe, J. (1991). Joint spacing in sedimentary rocks. *Journal of Structural Geology*, 13:1037–1048.

Nelson (2001). *Geologic Analysis of Naturally Fractured Reservoirs*. Butterworth-Heinemann, 2nd ed. edition.

Neufeld, C. and Wilde, B. (2005). A global kriging program for artifact-free maps. In *Annual Report Seven*. Centre for Computational Geostatistics, University of Alberta.

Oda, M. (1985). Permeability tensor for discontinuous rock masses. *Geotechnique*, 35(4):483–495.

Odling, N., Gillespie, P., Bourgine, B., Castaing, C., Chiles, J. P., Christensen, N., Fillion, E., Genter, A., Olsen, L., Thrane, R., Trice, R., Aarseth, E., and Walsh, J. (1999). Variations in fracture systems geometry and their implications for fluid flow in fractured reservoirs. *Petro*, 5:373–384.

Olarewaju, J., Ghori, S., Fuseni, A., and Wajid, M. (1997). Stochastic simulation of fracture density for permeability field estimation. Bahrain. Society of Petroleum Engineers, SPE 37962.

Ozkaya, S. (2003). Fracture length estimation from borehole image logs. *Mathematical Geology*, 35(6):737–753.

Pana, D. I., Waters, J., and Grobe, M. (2001). GIS compilation of structural elements in Northern Alberta, release 1.0. Technical Report 2001-01, Alberta Energy and Utilities Board - Alberta Geological Survey.

Park, R. (2004). *Foundations of Structural Geology*. Routledge.

Pearson, K. (1920). Notes on the history of correlation. *Biometrika*, 13(1):25–45.

Penny, K. I. and Jolliffe, I. T. (2001). A comparison of multivariate outlier detection methods for clinical laboratory safety data. *The Statistician*, 50(3):295–308.

Price, N. (1966). *Fault and Joint Development in Brittle and Semibrittle Rock*. Pergamon Press.

Priest, S. and Hudson, J. (1976). Discontinuity spacings in rock. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 13(5):135–148.

Prism Seismic Inc. (2012). REFRACT: integrated fractured reservoir modeling. Software.

Pyrak-Nolte, L., Montemagno, C., and D.D., N. (1997). Volumetric imaging of aperture distributions in connected fracture networks. *Geophys. Res. Letters*, 24(18):2343–2346.

Renshaw, C. and Pollard, D. (1994). Numerical simulation of fracture set formation: a fracture mechanics model consisted with experimental observations. *Journal of Geophys. Res.*, 99(9):372.

Reza, Z. A. (2003). *Some Aspects of Production Data Integration in Reservoir Modeling.* Ph.D. thesis, University of Alberta, Edmonton, Alberta.

RMOTC (2009). *Teapot Dome Data Set.* Rocky Mountain Oil Field Testing Center; U.S. Department of Energy.

Rodgers, J. L. and Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66.

Rodriguez, A. A., Klie, H., Sun, S., Gai, X., Wheeler, M. F., and Florez, H. (2006). Upscaling of hydraulic properties of fractured porpous media: Full permeability tensor and continuum scale simulations. In *SPE/DOE Symposium on Improved Oil Recovery, SPE 100057.*

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880.

Rousseeuw, P. J. and Zomeren, B. C. v. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411):633–639.

Roxar (2009). *Petrophysics Modeling.* Roxar.

Schlumberger (2007). *Fracture Modelling Course (Petrel 2007 Fracture Modelling Course Textbook).* Schlumberger.

Schlumberger (2009). *Petrel Discrete Fracture Modeling*, volume 2009.

Schwartz, B. (2006). Fracture pattern characterization of the Tensleep Formation, Teapot Dome, Wyoming. M.Sc. thesis, West Virginia University, West Virginia.

Shevlyakov, G. L. (1997). On robust estimation of a correlation coefficient. *Journal of Mathematical Sciences*, 83(3):434–438.

Smith, V. (2008). Modeling natural fracture networks: Establishing the groundwork for flow simulation at Teapot Dome, Wyoming. M.Sc. thesis, West Virginia University, West Virginia.

Srivastava, R. M. (2006). Field verification of a geostatistical method for simulating fracture network models. In *The 41st U.S. Symposium on Rock Mechanics (USRMS): 50 Years of Rock Mechanics - Landmarks and Future Challenges.*

Stach, S., Cybo, J., and Chmiela, J. (2001). Fracture surface: fractal or multifractal? *Materials Characterization*, 46(2-3):163–167.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*, volume 545. John Wiley and Sons, Inc.

Terzaghi, R. (1965). Sources of error in joint surveys. *Geotechnique*, 15(3):287–303.

Titterington, D. M. (1978). Estimation of correlation coefficients by ellipsoidal trimming. *Applied Statistics*, 27(3):227–234.

Tran, N. H., Chen, Z., and Rahman, S. S. (2006). Integrated conditional global optimisation for discrete fracture network modeling. *Computers & Geosciences*, 32:17–27. Elsevier Ltd.

Twiss, R. and Moores, E. (1992). *Structural Geology.* Freeman and Company, New York.

U.S. Department of the Interior Bureau of Reclamation (1998). *Engineering Geology Field Manual.* U.S. Department of the Interior Bureau of Reclamation.

Verga, N., Jackson, J., and McKenzie, D. (2000). Detection and characterization of fractures in naturally fractured reservoirs. In *SPE Annual Technical Conference and Exhibition, SPE 63266*, Dallas.

Wackernagel, H. (2003). *Multivariate Geostatistics.* Springer.

Waldren, D. and Corrigan, A. F. (1985). An engineering and geological review of the problems encountered in simulating naturally fractured reservoirs. In *SPE Middle East Oil Technical Conference and Exhibition*, pages 311–316.

Wang, X. (2005). *Stereological Interpretation of Rock Fracture Traces on Borehole Walls and Other Cylindrical Surfaces.* PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Warburton, P. (1980). A stereological interpretation of joint trace data. *International Journal of Rock Mechanics and Mining Science & Geomechanics Abstracts*, 17(4):181–190.

Wines, D. and Lilly, P. (2002). Measurement and analysis of rock mass discontinuity spacing and frequency in part of the Fimiston open pit operation in Kalgoorlie, Western Australia: a case study. *International Journal of Rock Mechanics and Mining Science*, 39:589–602.

Witherspoon, P., Wang, J., Iwai, K., and Gale, J. (1980). Validity of cubic law for fluid flow in a deformable rock fracture. *Water Resources Research*, 16(6):1016–1024.

Wu, H. and Pollard, D. (1995). An experimental study of the relationship between joint spacing and layer thickness. *Journal of Structural Geology*, 17(6):887–905.

Xu, C. and Dowd, P. (2010). A new computer code for discrete fracture network modelling. *Computers & Geosciences*, 36:292–301.

Zhang, L., Einstein, H. H., and Dershowitz, W. S. (2002). Stereological relationship between trace length and size distribution of elliptical discontinuities. *Geotechnique*, 52(6):419–433.

Zielinski, K. and Laur, R. (2005). Stopping criteria for single-objective optimization. In *Proceednigs of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore.

Appendix A

Additional Material from the Teapot Dome Case Study

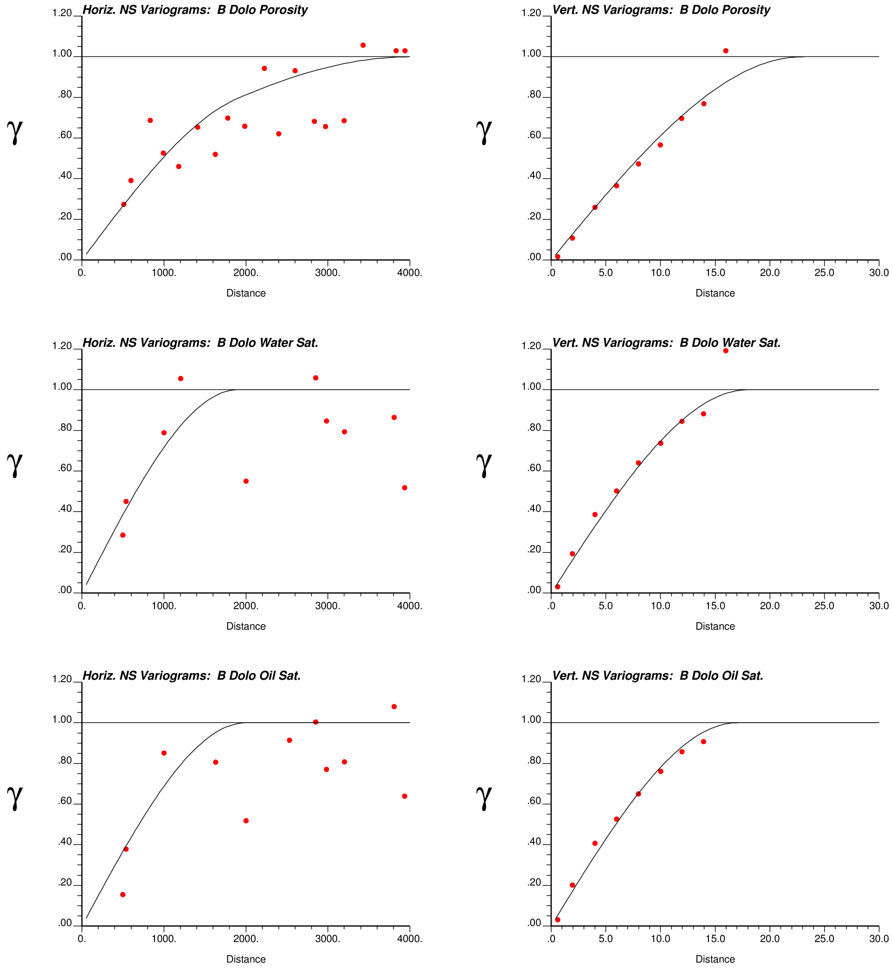


Figure A.1: Horizontal and vertical normal score semi-variograms for the B Dolomite Unit (top row = porosity, middle row = water saturation, bottom row = oil saturation).

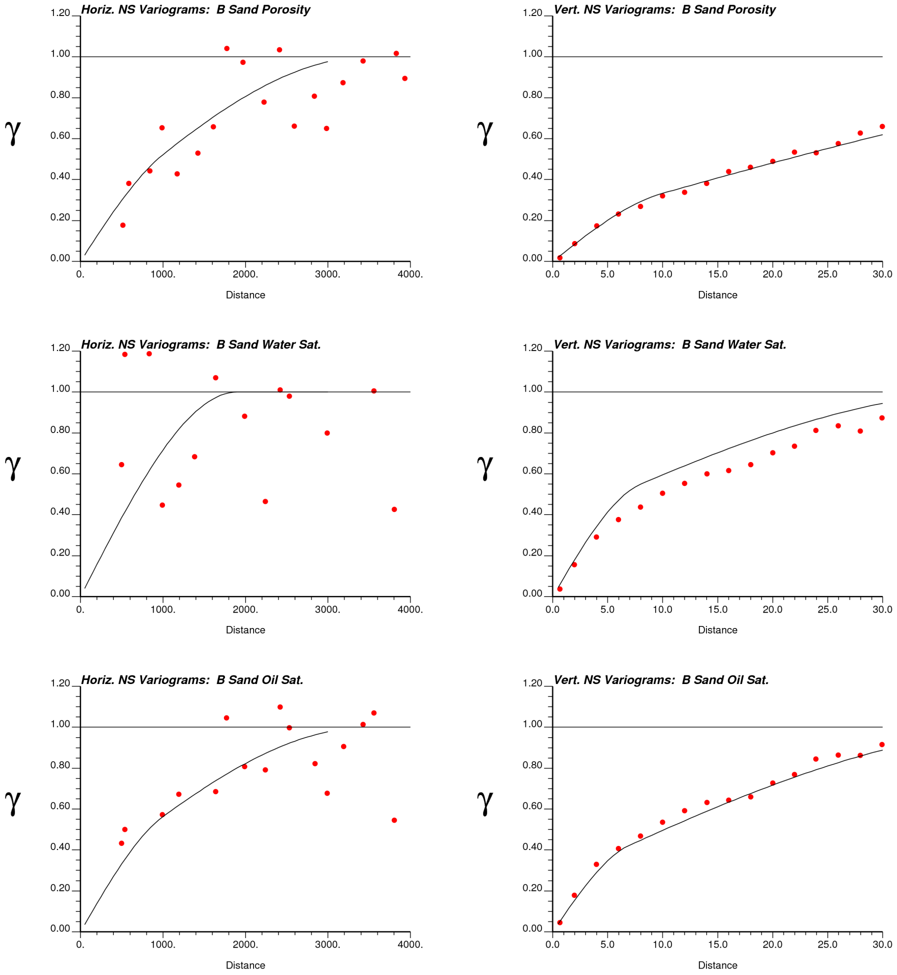


Figure A.2: Horizontal and vertical normal score semi-variograms for the B Sand Unit (top row = porosity, middle row = water saturation, bottom row = oil saturation).

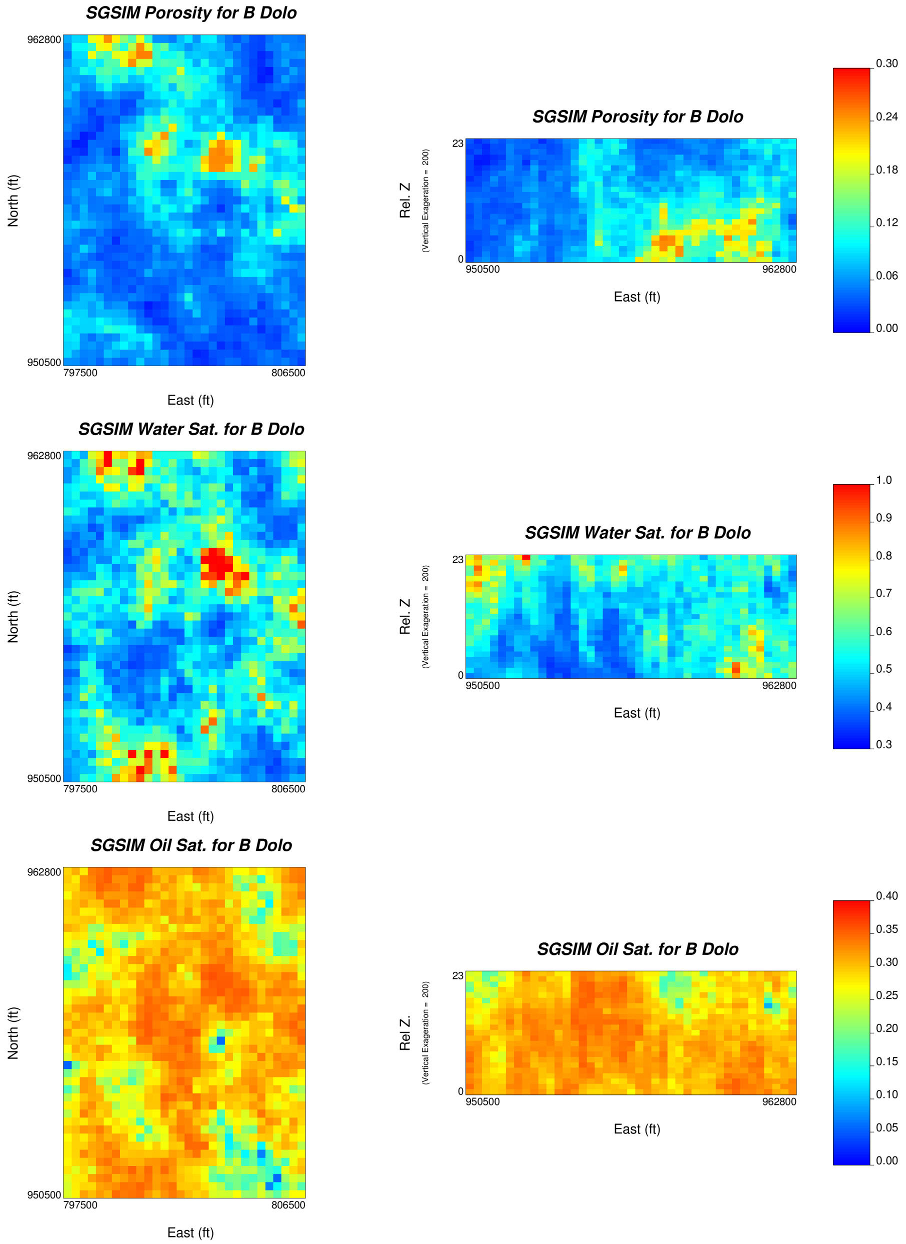


Figure A.3: Cross sections of one realization of porosity, water saturation and oil saturation through the B Dolomite Unit.

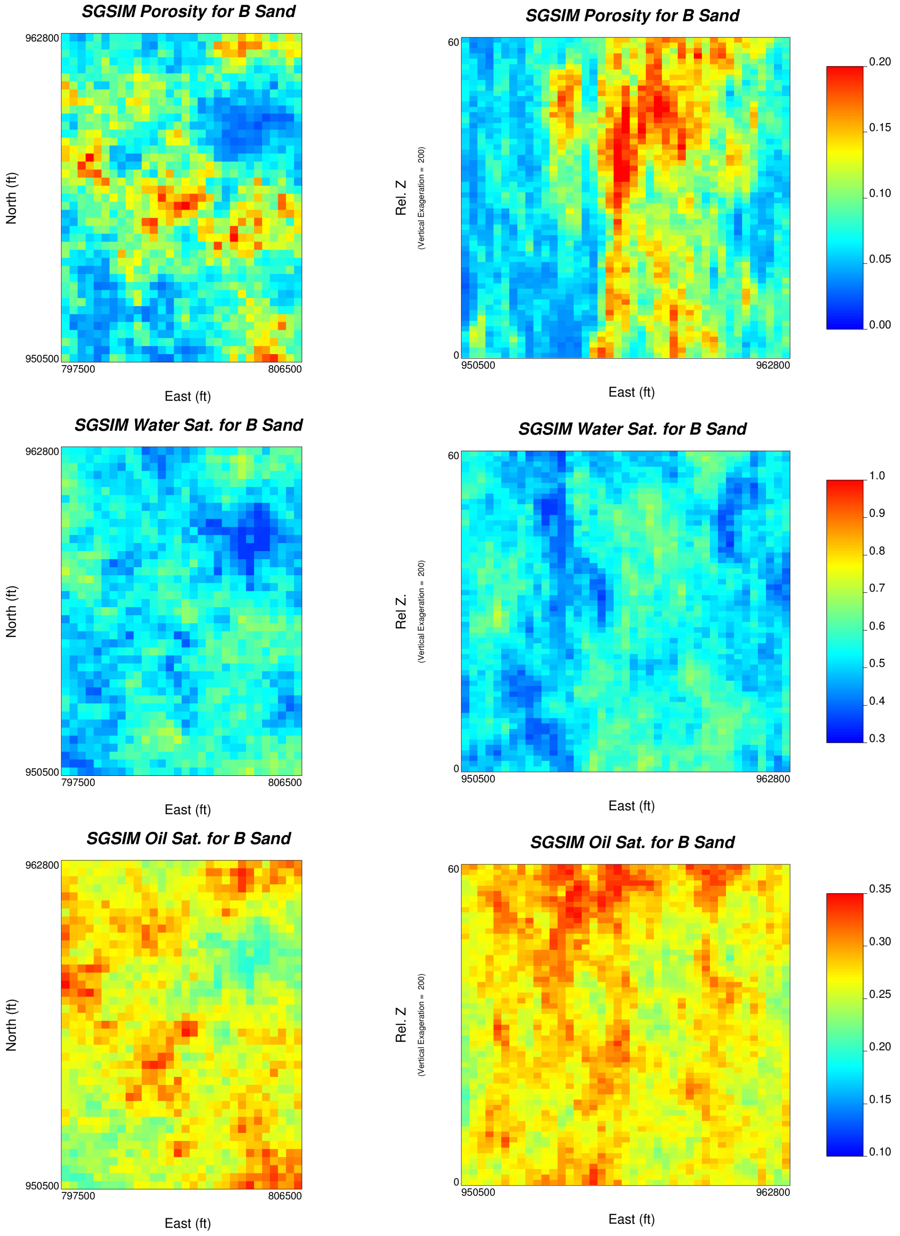


Figure A.4: Cross sections of one realization of porosity, water saturation and oil saturation through the B Sandstone Unit.

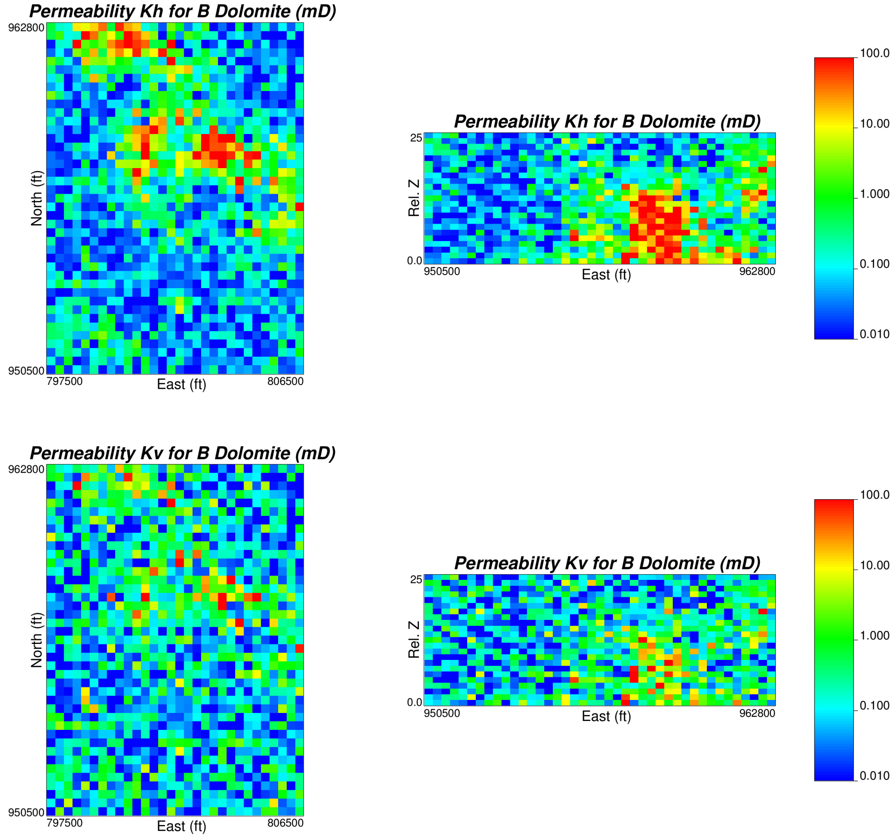


Figure A.5: Cross sections of one realization of horizontal and vertical permeability in the B Dolomite Unit.

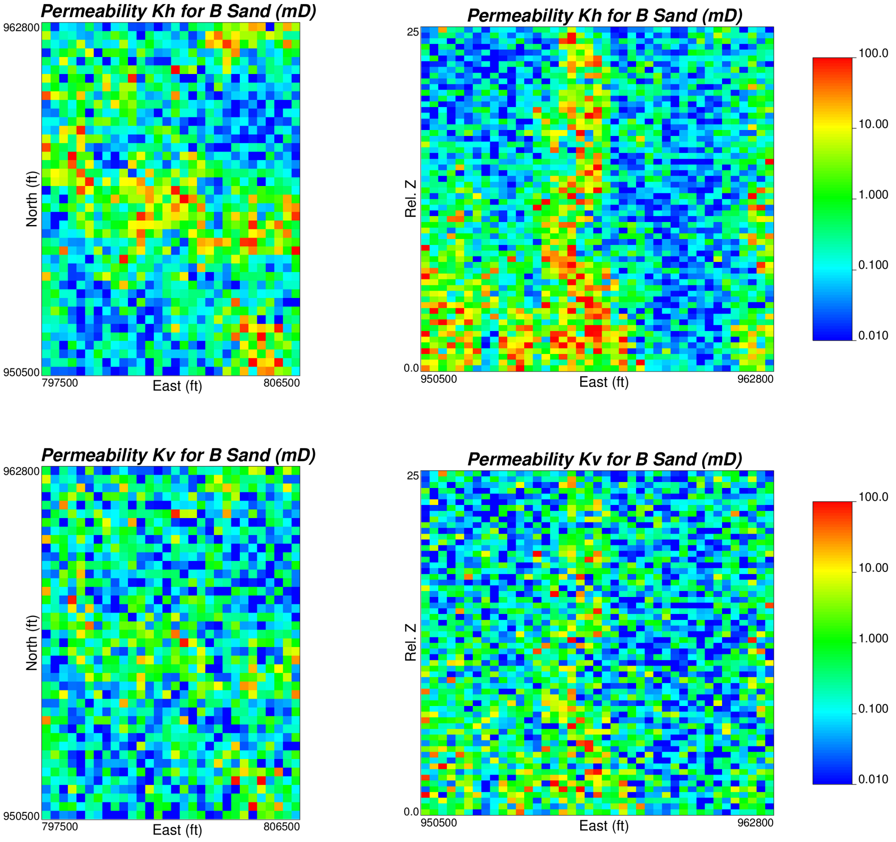


Figure A.6: Cross sections of one realization of horizontal and vertical permeability in the B Sand Unit.

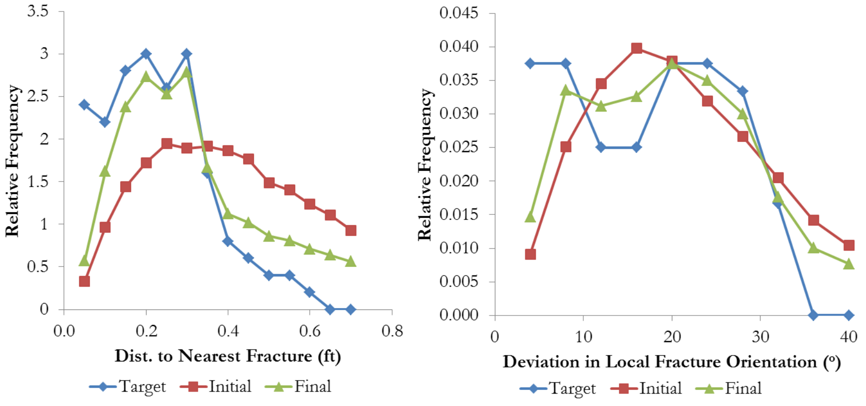


Figure A.7: Target, initial DFN and final DFN histograms for well 48-X-28 in the A Sandstone unit.

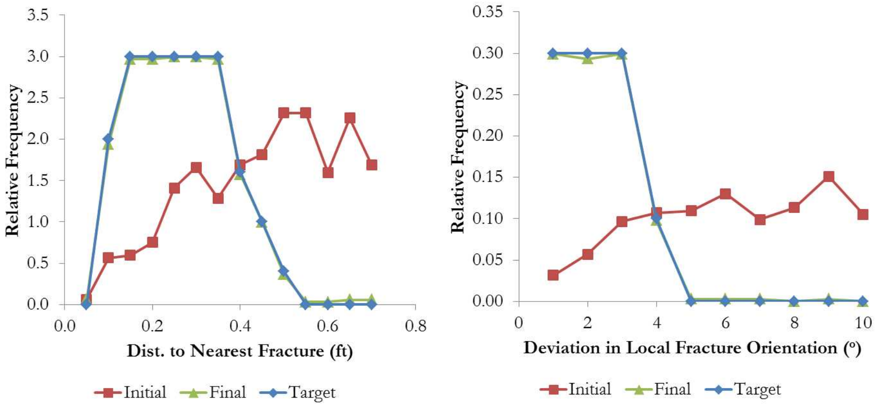


Figure A.8: Target, initial DFN and final DFN histograms for well 61-2-X-15 in the A Sandstone unit.

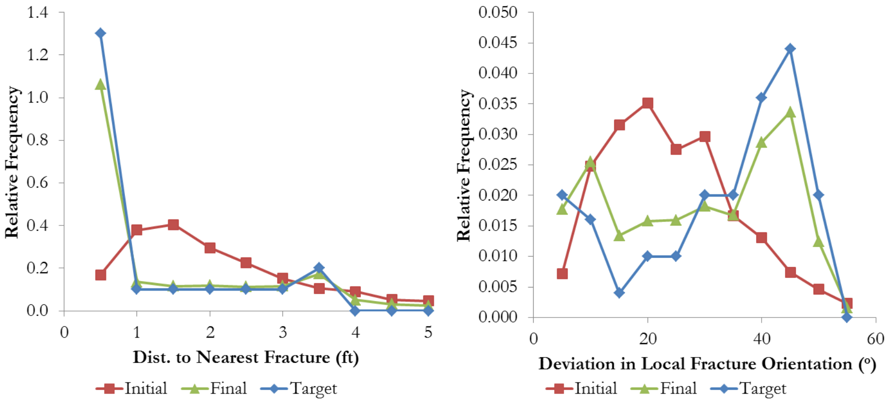


Figure A.9: Target, initial DFN and final DFN histograms for well 67-1-X-10 in the A Sandstone unit.

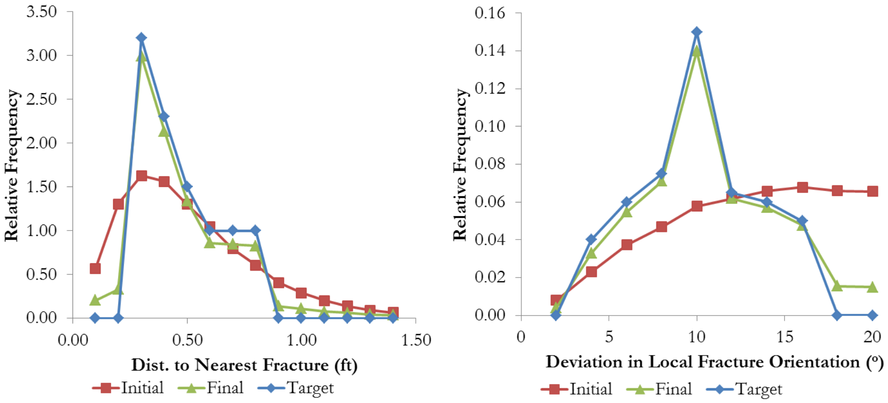


Figure A.10: Target, initial DFN and final DFN histograms for well 71-1-X-4 in the A Sandstone unit.

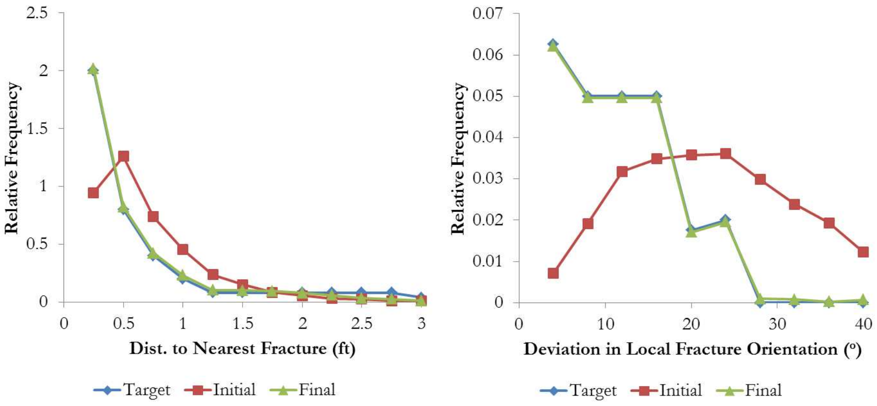


Figure A.11: Target, initial DFN and final DFN histograms for well 25-1-X-14 in the B Dolomite unit.

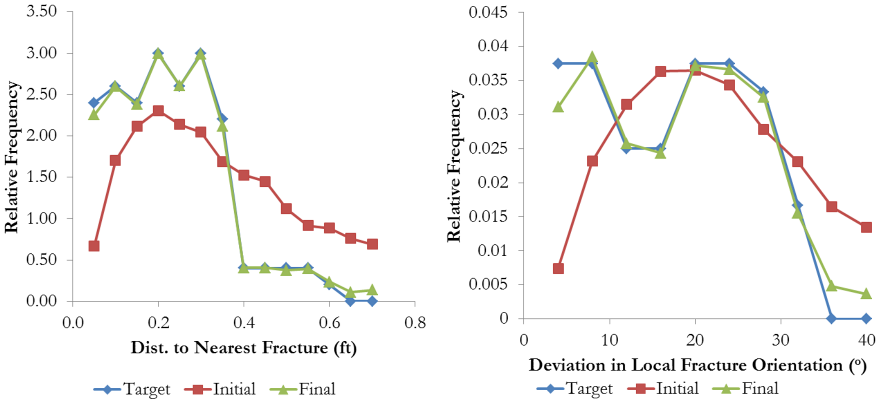


Figure A.12: Target, initial DFN and final DFN histograms for well 48-X-28 in the B Dolomite unit.

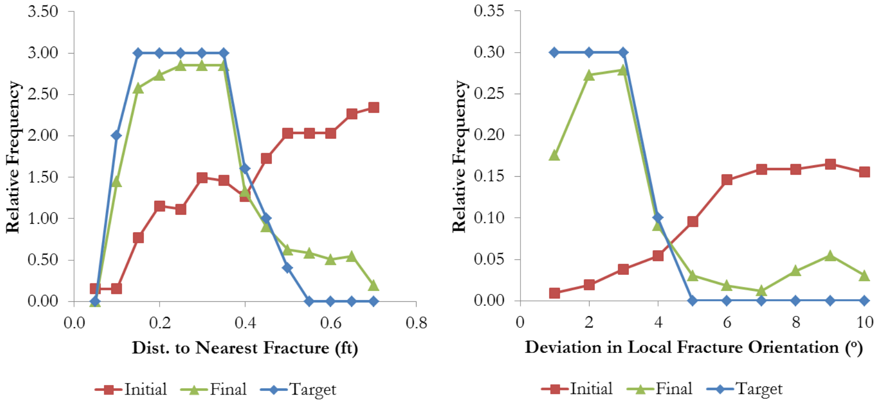


Figure A.13: Target, initial DFN and final DFN histograms for well 61-2-X-15 in the B Dolomite unit.

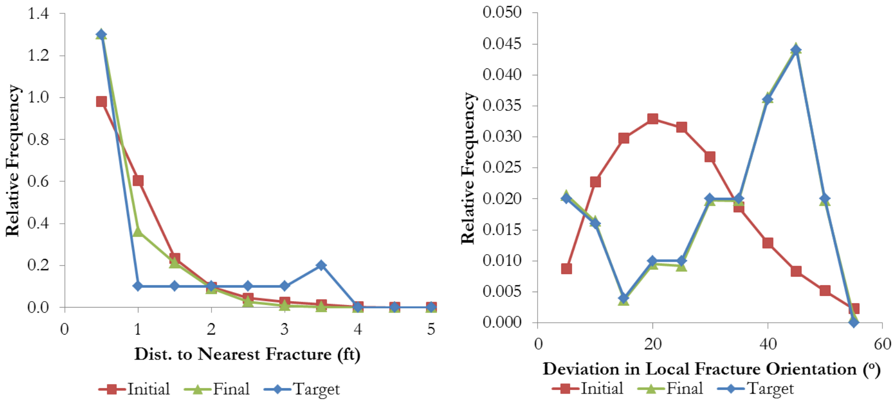


Figure A.14: Target, initial DFN and final DFN histograms for well 67-1-X-10 in the B Dolomite unit.

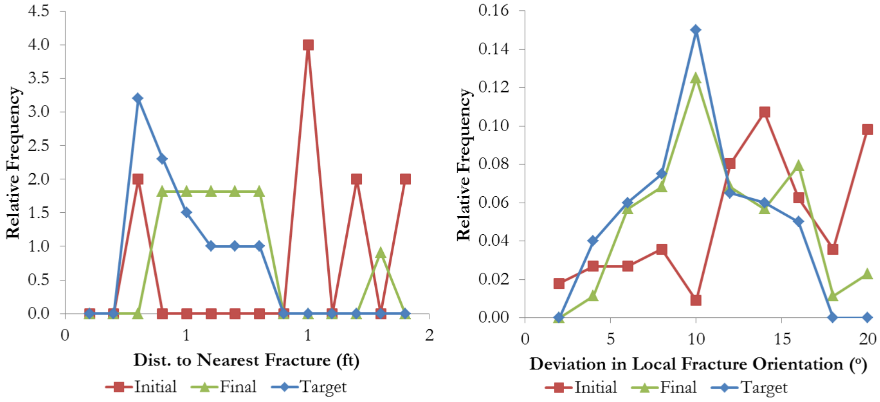


Figure A.15: Target, initial DFN and final DFN histograms for well 71-1-X-4 in the B Dolomite unit.

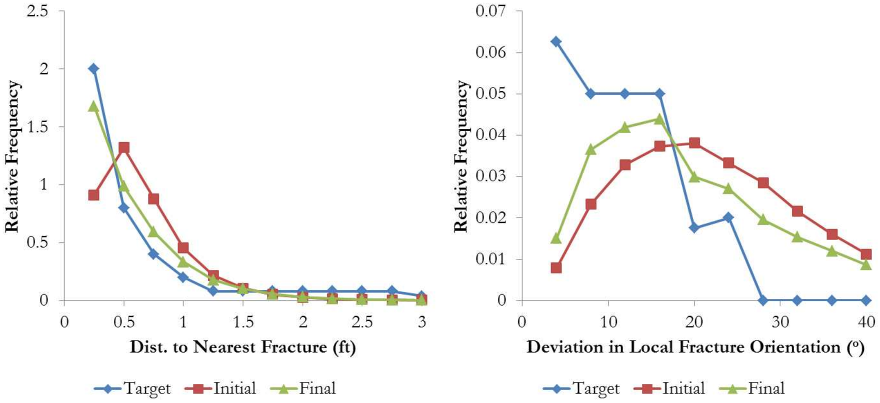


Figure A.16: Target, initial DFN and final DFN histograms for well 25-1-X-14 in the B Sandstone unit.

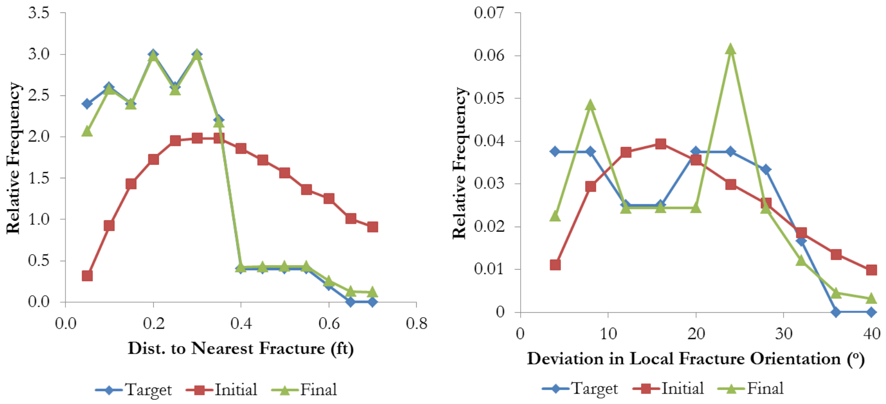


Figure A.17: Target, initial DFN and final DFN histograms for well 48-X-28 in the B Sandstone unit.

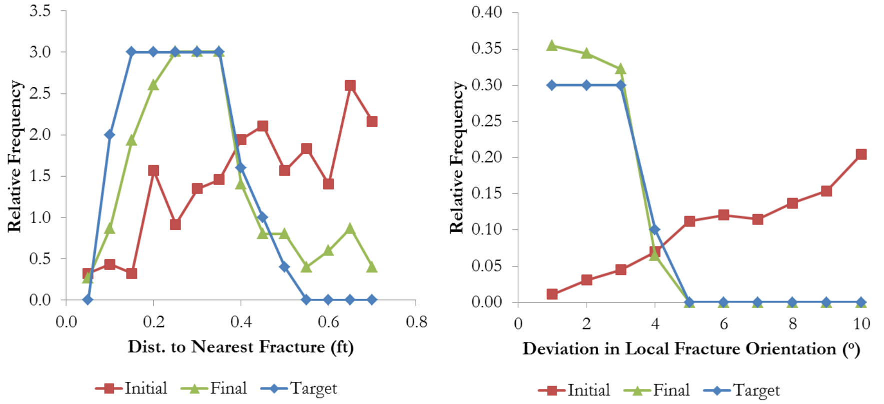


Figure A.18: Target, initial DFN and final DFN histograms for well 61-2-X-15 in the B Sandstone unit.

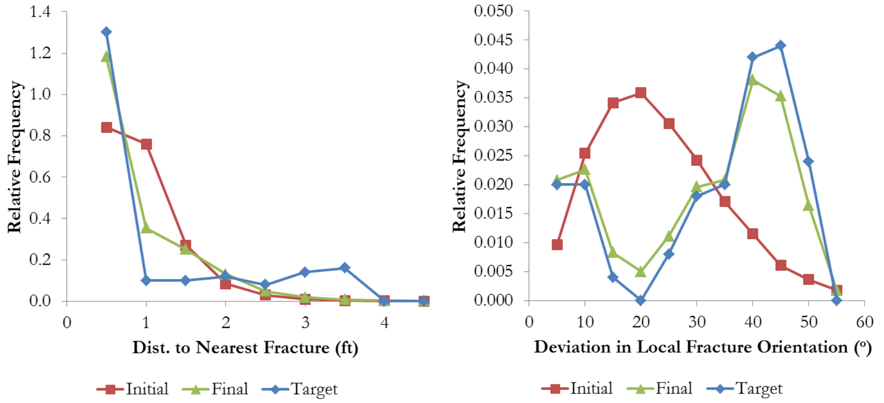


Figure A.19: Target, initial DFN and final DFN histograms for well 67-1-X-10 in the B Sandstone unit.

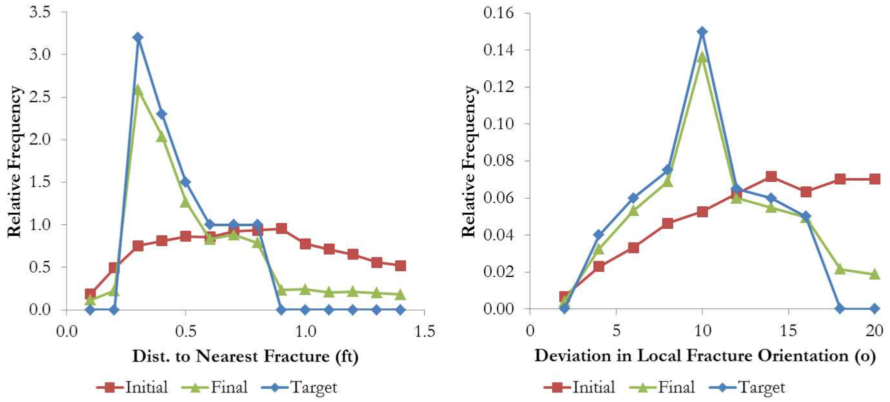


Figure A.20: Target, initial DFN and final DFN histograms for well 71-1-X-4 in the B Sandstone unit.

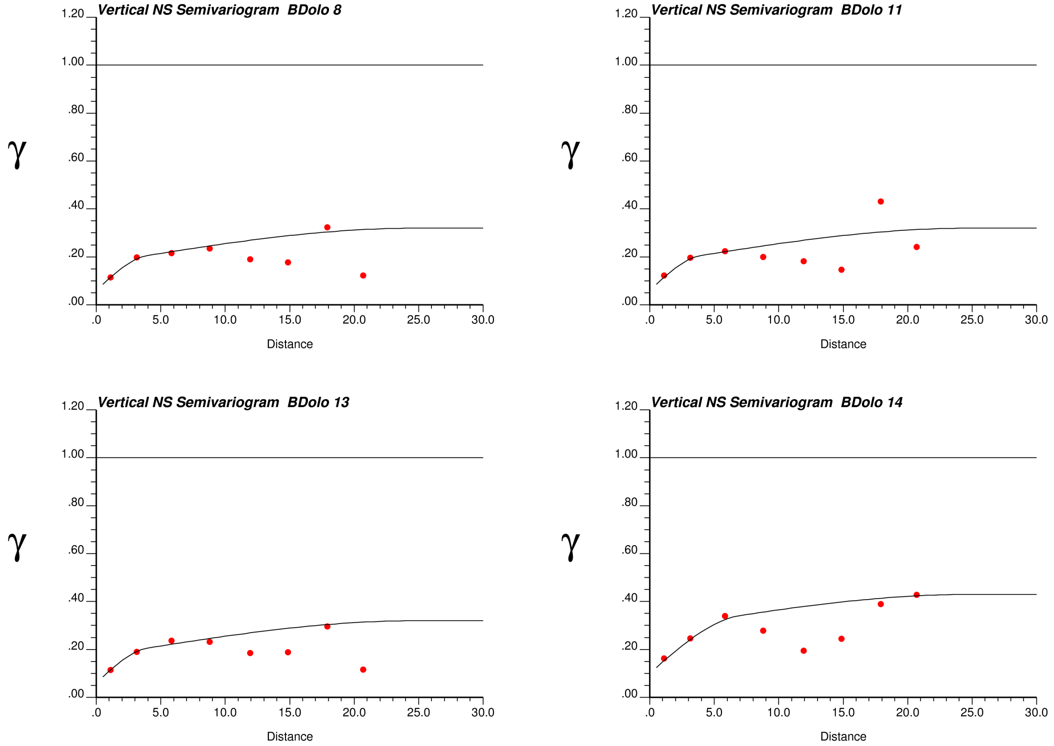


Figure A.21: Vertical semi-variograms for K_{xx} (top left), K_{yy} (top right) and K_{zz} (bottom left) and fracture porosity (bottom right) in the B Sandstone unit.

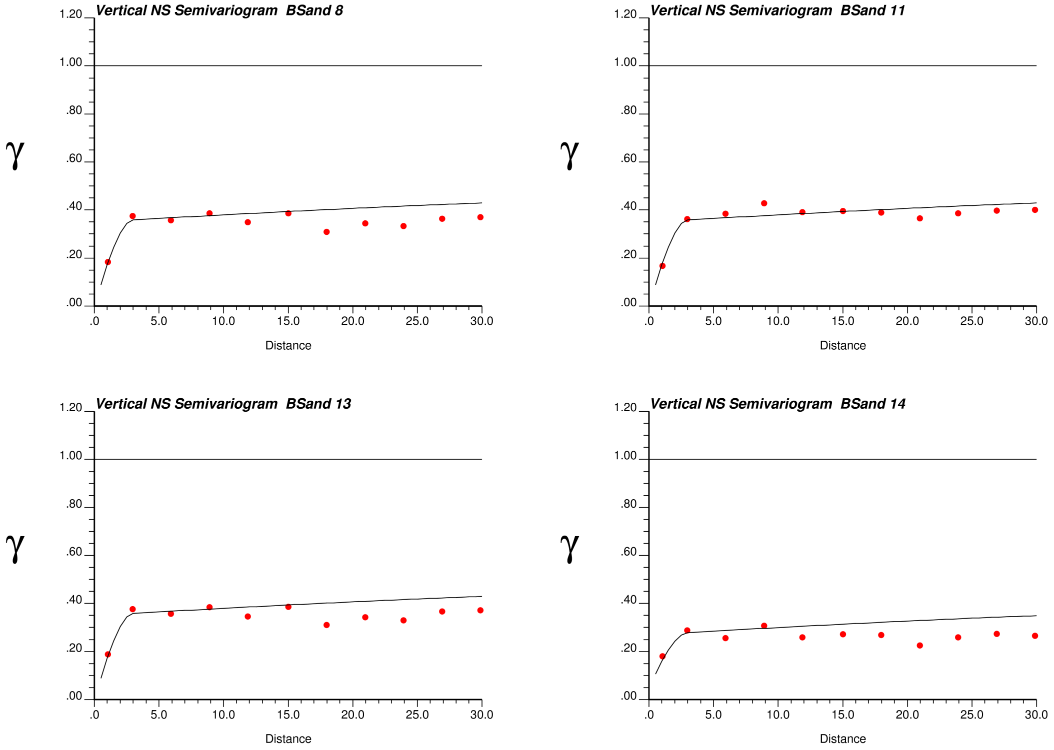
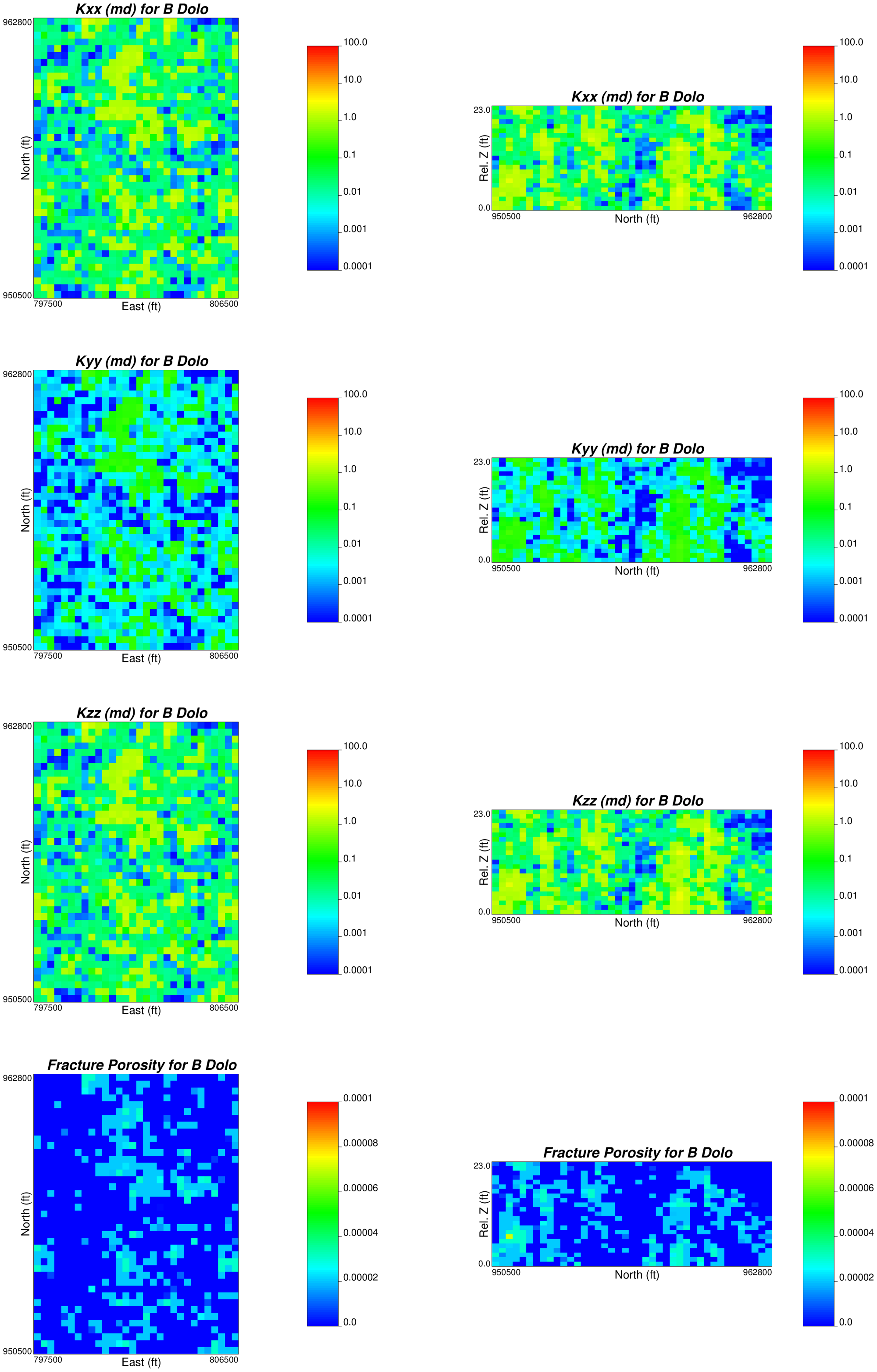


Figure A.22: Vertical semi-variograms for K_{xx} (top left), K_{yy} (top right) and K_{zz} (bottom left) and fracture porosity (bottom right) in the B Sandstone unit.

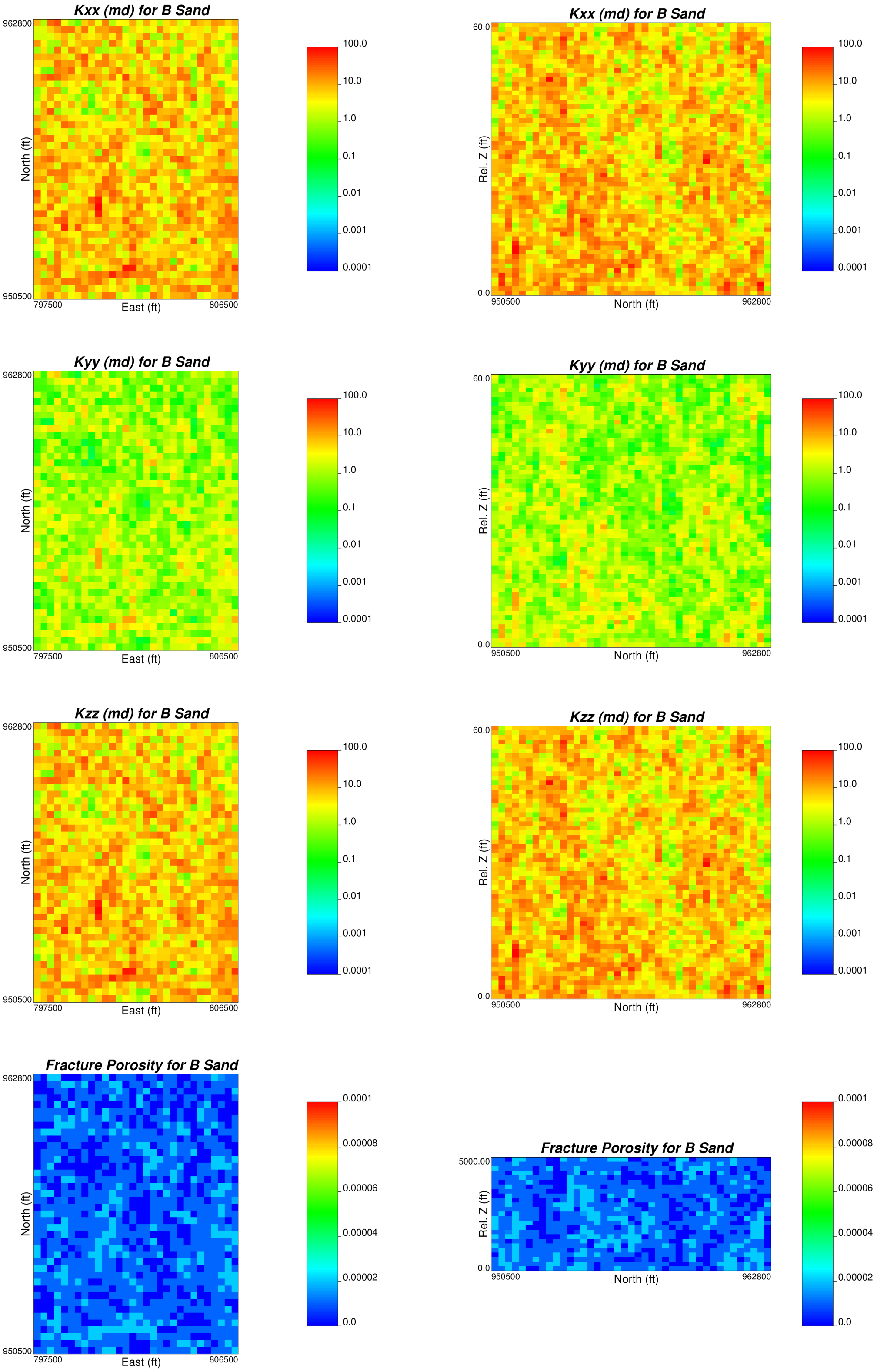

Kxx (md) for B Dolo

Kyy (md) for B Dolo

Kzz (md) for B Dolo

Fracture Porosity for B Dolo

Figure A.23: One realization of the simulated K_{xx}, K_{yy}, K_{zz} and fracture porosity in the B Dolomite unit.



Kxx (md) for B Sand

Kyy (md) for B Sand

Kzz (md) for B Sand

Fracture Porosity for B Sand

Figure A.24: One realization of the simulated K_{xx}, K_{yy}, K_{zz} and fracture porosity in the B Sandstone unit.