

A New Approach to DFN Simulation

Eric B. Niven and Clayton V. Deutsch

Discrete Fracture Networks (DFNs) are created to model natural fractures and understand their effect on engineering design. Traditional methods for creating DFNs generate fracture locations (for each set of fractures) randomly and simulate orientation from user-defined distributions. DFNs created in this manner do not explicitly honor fracture spacing and local fracture orientation. This results in fracture networks that may not have realistic spatial properties, such as fractures that are too close or too far apart. Moreover, unlikely scenarios including fractures from the same set crossing at low angles may also be observed. This paper presents a new approach to DFN simulation that relies on generating more fractures than are required and iteratively adding or removing fractures to find a subset that matches the input distributions of fracture spacing, orientation and intensity. Unlikely scenarios, such as crossings, are avoided.

1. Introduction

A DFN is a population of individual fractures that work together as a single hydrological and/or mechanical system. DFNs have a huge effect on fluid flow and the production of fractured reservoirs. Most commercially available DFN computer codes use a Poisson or a non-homogeneous Poisson process to generate fracture locations randomly (Cacas et al. 2001, Gauthier et al. 2002, Hitchmough et al. 2007). These codes also simulate fracture orientation independent of location. This process may lead to fractures from the same joint set that are unrealistically close together, unrealistically far apart or that cross at very low angles (see Niven and Deutsch (2010b) for more discussion on this matter).

This paper is part of a series of four articles in this report that pertain to fracture modeling. Niven and Deutsch (2010b) discuss and present evidence that some natural fracture networks cannot be accurately modeled using traditionally created DFNs. This paper discusses the proposed approach and methodology to DFN simulation. Niven and Deutsch (2010c) present a new computer program to simulate DFNs using the methodology described in this paper. Finally, Niven and Deutsch (2010a) present an example of the program and methodology applied to a fracture modeling problem from Northern Alberta.

2. Generating the Initial DFN

Under the proposed approach, the first step is to generate a large pool of fractures. The number of fractures generated depends on the user defined *fracture multiplication factor*. The fracture multiplication factor is the ratio of the number of fractures in the pool to the target number of fractures in the final DFN. Thus, if the target number of fractures is 10 and the fracture multiplication factor is 2, 20 fractures are generated for the pool of fractures. This requires a priori knowledge of the desired number of fractures in the final DFN model. Using a higher multiplication factor results in better DFN models but requires additional CPU time. Usually, a multiplication factor between 2 and 5 is sufficient, as this generates enough fractures such that no areas are left without fractures.

The pool of fractures is split into two groups as shown in Figure 1. Fractures that are in the activated group make up the initial DFN. The number of fractures in the initial DFN equals the target fracture intensity. The initial DFN represents a starting DFN that is iterated upon in order to arrive at a better DFN (as measured by an objective function, which is discussed later). Those fractures not assigned to the activated group are assigned to the deactivated group.

The steps in generating the pool of fractures are:

1. First a grid cell is visited. The centroid of a fracture is randomly generated in the grid block.
2. A fracture is specified as a rectangular plane whose orientation can be defined by the orientation of its pole. The orientation of a fracture pole is defined by its trend and plunge. Once the fracture centroid location has been simulated, the fracture orientation is simulated from a specified input distribution.
3. Fracture length, height and aperture are simulated from their input distributions.
4. This process repeats until the pool holds the required number of fractures. The required number of fractures equals the target fracture intensity multiplied by the fracture multiplication factor.
5. The target number of fractures for the DFN is activated. Any fractures not activated are considered to be deactivated. The fractures that are activated in this step constitute the *initial DFN*.

3. Calculating the Fracture Spacing of a DFN

After the pool of fractures and the initial DFN are created, a measure of fracture spacing for the DFN is determined. However, there are multiple approaches for determining fracture spacing.

Calculating “True” Fracture Spacing

A geologist or engineer might wish to measure fractures exposed in an outcrop of rock. The geologist might set up a scan line and measure the location and orientation of any fractures that intersect the scan line. The fracture spacing, measured as the distance between fractures on the scan line, could be calculated.

A program called CALCSPACING was created to calculate the distribution of fracture spacing for a DFN. The program works by generating a random location and projecting a scan line using the mean fracture pole trend and plunge. The points where the scan line intersects the fractures are noted, and the fracture spacing is calculated as the distance between the intersection points. Figure 2 illustrates this. The user specifies the number of scan lines to use ensuring that each fracture is sampled many times and that a stable distribution of fracture spacing is built.

Calculating the Perpendicular Distance to the Nearest Fracture

The above mentioned true fracture spacing calculation is computationally expensive when it needs to be repeated many times. The proposed methodology for simulating DFNs uses the average perpendicular distance between each fracture and its nearest neighbor.

Each fracture is visited and the nearest fracture is identified. Fracture spacing is normally measured normal to the plane of the fracture. Thus, the average perpendicular distance between fractures is used (Figure 3). The perpendicular distances (d_{p2} and d_{p1}) between each fracture’s centroid and the other fracture are determined. The average of d_{p2} and d_{p1} is taken to be the fracture spacing. If the ray from one centroid does not intersect the fracture, the intersection is taken where it would have been if the fracture was infinite in extents. Note that the program CALCSPACING also calculates the distribution of the perpendicular distance to the nearest fracture, in addition to the true fracture spacing distribution.

Figure 4 illustrates the geometry to determine the perpendicular distance between the fractures. First, the normal vector to the fracture, $\mathbf{n} = [a, b, c]$, is determined:

$$a = \sin(p_{trend}) \cos(p_{plunge}) \quad (1)$$

$$b = \cos(p_{trend}) \cos(p_{plunge}) \quad (2)$$

$$c = \sin(p_{plunge}) \quad (3)$$

where p_{trend} and p_{plunge} are the trend and plunge of the fracture’s pole, measured counterclockwise from north and below the horizontal plane, respectively. The equation for a plane that represents the fracture is:

$$ax + by + cz + d = 0 \quad (4)$$

The fracture centroid is a point $\mathbf{p}_0 = (x_0, y_0, z_0)$. Thus, a vector from the plane to the point, \mathbf{p}_0 , is given by:

$$\mathbf{w} = - \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} \quad (5)$$

Projecting \mathbf{w} onto \mathbf{n} gives the distance, D , from the point to the plane as:

$$D = |\text{proj}_{\mathbf{n}} \mathbf{w}| \quad (6)$$

$$= \frac{|\mathbf{n} \cdot \mathbf{w}|}{|\mathbf{n}|} \quad (7)$$

$$= \frac{|a(x - x_0) + b(y - y_0) + c(z - z_0)|}{\sqrt{a^2 + b^2 + c^2}} \quad (8)$$

$$= \frac{|ax - ax_0 + by - by_0 + cz - cz_0|}{\sqrt{a^2 + b^2 + c^2}} \quad (9)$$

$$= \frac{|-d - ax_0 - by_0 - cz_0|}{\sqrt{a^2 + b^2 + c^2}} \quad (10)$$

Thus, the perpendicular distance from the fracture centroid, \mathbf{p}_0 , to the plane is:

$$d_{p1} = D = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (11)$$

The fractures are assumed to be infinite in length and height. This allows calculation of the perpendicular distance even if the fractures are oriented differently or are skewed in location as illustrated in Figure 3.

4. Calculating the Nearest Fracture Inter-Angle

In addition to the fracture spacing, the similarity in orientation between the nearest fractures is compared. The similarity in orientation is determined by finding the angle between the two fracture poles. This angle is termed the *nearest fracture inter-angle*.

The nearest fracture inter-angle is calculated from the dot product of the two normal vectors as follows:

$$\theta_{\text{inter-angle}} = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right) \quad (12)$$

where \mathbf{a} and \mathbf{b} are the normal vectors to two fractures, \mathbf{a} and \mathbf{b} , and fracture \mathbf{b} is the nearest fracture to \mathbf{a} .

5. The Objective Function

An objective function is used to calculate the match between the input spacing, inter-angle and intensity distributions and those statistics from the activated DFN. There are three components to the objective function; a fracture spacing objective component, an inter-angle component and a fracture intensity component. The objective function is:

$$\text{Obj. Fn.} = \sum_{i=1}^{sbins} (S_{\text{input},i} - S_{\text{activated},i})^2 + \sum_{i=1}^{abins} (A_{\text{input},i} - A_{\text{activated},i})^2 + \eta \times \sum_{i=1}^{ibins} (I_{\text{input},i} - I_{\text{activated},i})^2 \quad (13)$$

where:

- $S_{\text{input},i}$ and $S_{\text{activated},i}$ are the target (input) fracture spacing distribution and the fracture spacing distribution of the activated DFN, respectively. $sbins$ is the number of bins in the PDF.
- $A_{\text{input},i}$ and $A_{\text{activated},i}$ are the target (input) nearest fracture inter-angle distribution and the nearest fracture inter-angle distribution of the activated DFN, respectively. $abins$ is the number of bins in the PDF.
- $I_{\text{input},i}$ and $I_{\text{activated},i}$ are the target (input) fracture intensity and the fracture intensity of the activated DFN, respectively. $ibins$ is the number of bins in the PDF.

- η is a constant that allows some flexibility in the intensity objective function. This parameter is discussed further in Niven and Deutsch (2010c).
- The three input distributions are non-parametric.

6. Activating and Deactivating Fractures

A greedy optimization algorithm is implemented. A fracture is randomly selected from the pool of all fractures and its activation state is changed. That is, if the fracture is already activated, that fracture is deactivated. If the fracture is already deactivated, that fracture is activated.

After switching the activation of the selected fracture, the objective function is re-calculated. If the activation change decreases the objective function, the change is kept. If the activation change increases the objective function, the change is rejected. The process repeats until the desired number of iterations are complete.

7. Conclusions

Traditional methods for generating DFNs do not attempt to honor distributions of fracture spacing and nearest-fracture orientation from measured data. A new method for generating DFNs was presented. The new method generates more fractures than are required and iterates to find a subset of fractures that best matches the target fracture spacing, orientation and intensity.

References

- Cacas, M.C., Daniel, J.M., and Letouzey, J. 2001. Nested geological modelling of naturally fractured reservoirs, *Petroleum Geoscience*, **7**(S): S43-S52.
- 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**: 284-294.
- Hitchmough, A.M., Riley, M.S., Herbert, A.W., and Tellam, J.H. 2007. Estimating the hydraulic properties of the fracture network in a sandstone aquifer, *Journal of contaminant hydrology*, **93**(1-4): 38-57.
- Niven, E.B. and Deutsch, C.V. 2010a. An example application of DFNSIM in modeling surface lineaments. Paper 206, Report 12, Centre for Computational Geostatistics, University of Alberta, Edmonton, Alberta.
- Niven, E.B. and Deutsch, C.V. 2010b. On the randomness of natural fractures. Paper 207, Report 12, Centre for Computational Geostatistics, University of Alberta, Edmonton, Alberta.
- Niven, E.B. and Deutsch, C.V. 2010c. DFNSIM: A new program for simulating discrete fracture networks. Paper 402, Report 12, Centre for Computational Geostatistics, University of Alberta, Edmonton, Alberta.

Figures

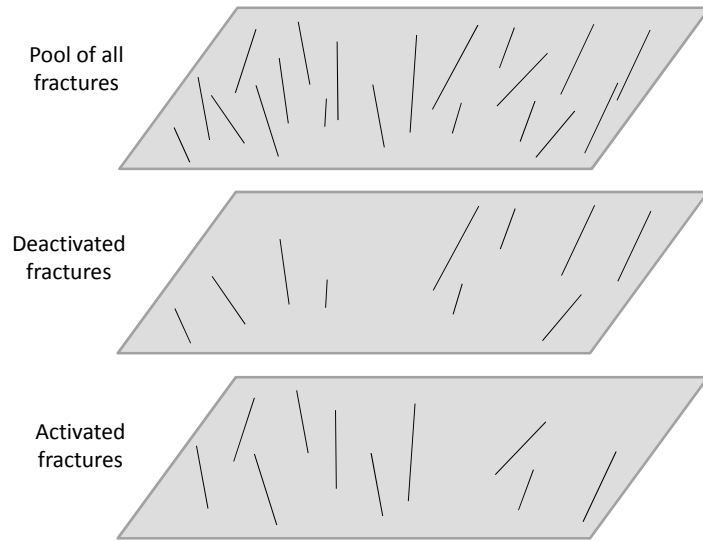


Figure 1: A 2D illustration of the activated, deactivated and pool of fractures. There are 10 activated and 10 deactivated fractures. Thus, there are 20 fractures in the pool. The activated fractures represent the DFN under construction. The deactivated fractures are thrown away when the desired number of iterations is complete.

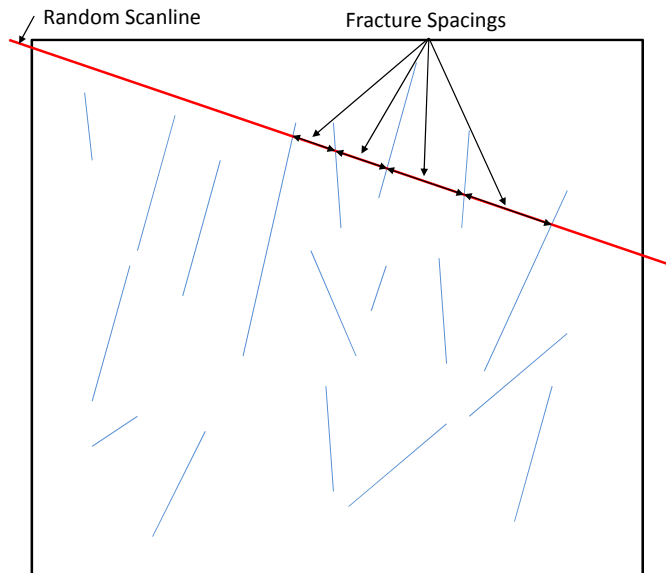


Figure 2: An illustration of the CALCSPACING program. The program projects a random scan line, that is parallel to the mean fracture pole (or, its normal vector), finds the intersections between the line and the fractures and calculates the spacing between the fractures.

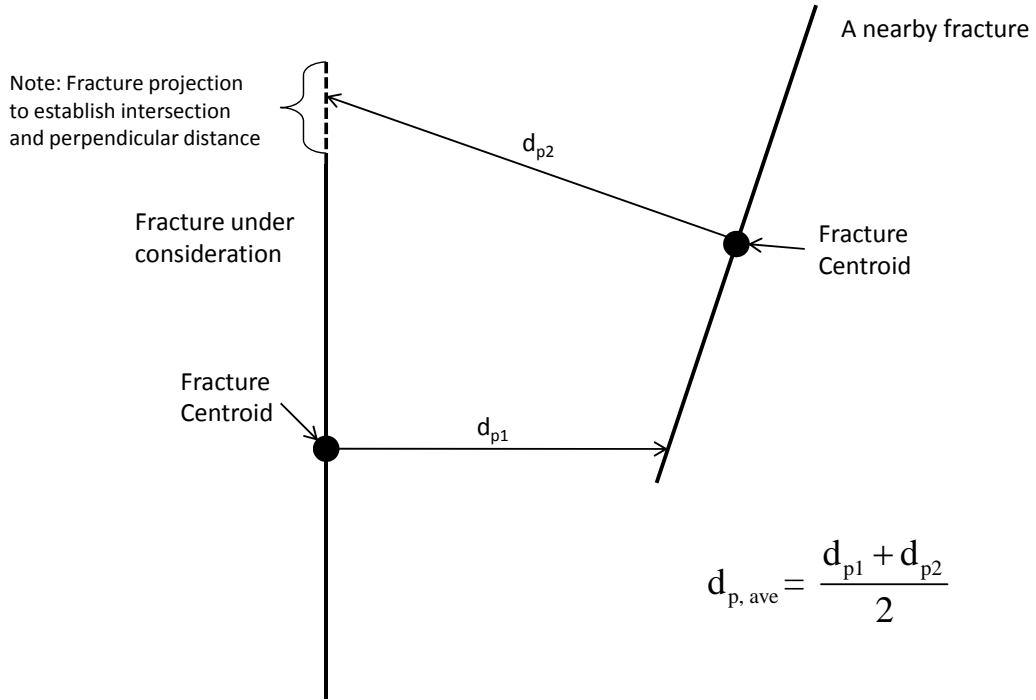


Figure 3: The procedure for calculating the perpendicular distance to the nearest fracture.

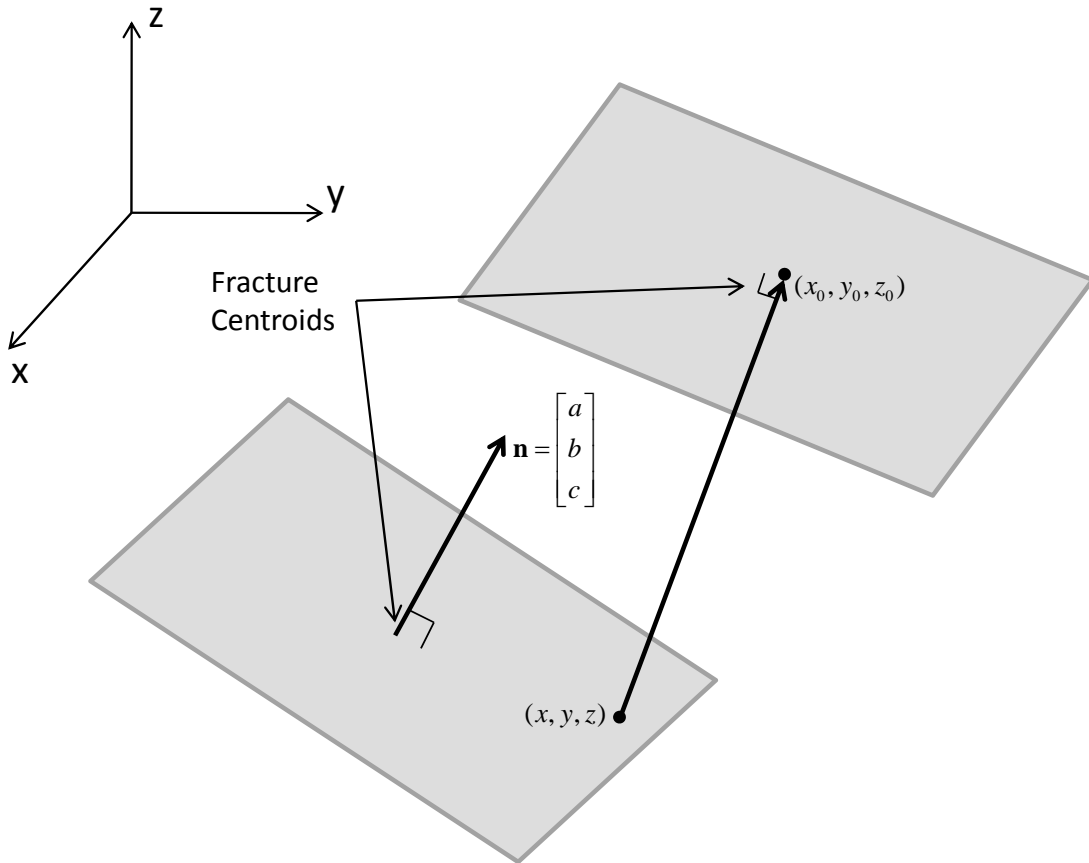


Figure 4: A diagram of the point-plane distance calculation.