An Example Application of DFNSIM in Modeling Surface Fractures

Eric B. Niven and Clayton V. Deutsch

This paper presents an application of `DFNSIM` for modeling surface fractures from Northern Alberta. The relevant statistics for the fracture data are presented. Then the modeling parameters are discussed in detail. Since `DFNSIM` outputs initial (unoptimized) and final (optimized) discrete fracture networks (DFNs), the improvement from the initial to the final DFN is examined. Qualitatively, the final (optimized) DFN look more similar to the fracture map than the initial DFN. This is supported by relevant statistics as it is shown that the fracture spacing for the initial DFN is a poor match for the target fracture spacing while the final DFN is a better match. Similarly, the initial DFN provides a poor match for fracture orientation while the final DFN is nearly identical to the target distribution. The final DFN also better matches the target value for fracture intersections compared to the initial DFN.

1. Introduction

DFNs are commonly generated to model the fractures in a reservoir. Large scale faults and some large scale fractures show up on seismic surveys and are explicitly modeled. Small and medium scale fractures are modeled probabilistically. DFNs have the advantage of portraying both the fracture spatial distribution and details of individual fracture characteristics such as location, orientation, size, density and conductivity (Tran et al. 2002). Most DFN generation methodologies and computer code assume the fractures are randomly distributed in space using a Poisson process (Cacas et al. 2001, Gauthier et al. 2002, Hitchmough et al. 2007).

This paper is part of a series of four articles in this report that pertain to fracture modeling. In this paper a real-life example is presented. `DFNSIM` is used to model measured surface fractures from Northern Alberta. The modeling process is discussed in detail. Niven and Deutsch (2010b) discuss and present evidence that some natural fracture networks cannot be accurately modeled using traditionally created DFNs. Niven and Deutsch (2010a) present the new approach to modeling DFNs. Niven and Deutsch (2010c) present a new computer program to simulate DFNs using the methodology described in this paper.

2. Example: Surface Fractures

Figure 1 shows a map of fractures from an area of Northern Alberta (Pana and Waters 2001). The figure shows 2 fracture sets, one trending southwest-northeast and the other trending southeast-northwest. The southwest-northeast fracture set was digitized for this example. In total, there are 425 fractures from the SW-NE set that were digitized.

Figure 1 also shows the digitized fracture set. Figure 2 and Figure 3 show a contoured stereonet and a rose diagram of the fractures, respectively. The fractures have an average pole trend of 146.5 with a standard deviation of 11.8 degrees. Although the stereonet and the rose diagrams indicate that the fractures are from a single set, from a qualitative standpoint there appears to be some pattern to the fracture locations and orientations. Namely, there are few fracture intersections and fractures appear to be oriented similarly to their nearest neighbors. While there is variation in the orientation of the fractures when looking at an area wide scale, locally fractures appear to be oriented similarly to their nearest neighbors.

3. The Parameter File for this Example

```
                  Parameters for DFNSIM
                  ***********************


START OF MAIN PARAMETERS:
1 503 1006                   -nx,xmn,xsiz
1 525 1050                   -ny,ymn,ysiz
1  0.05 0.1                  -nz,zmn,zsiz
6911                         -random number seed
----------------------------------------------------------------------
***Data Search Parameters***
1    1      1               -x,y and z block discretization
0                           -max per octant (0-> not used)
2000 2000 2000              -maximum search radii
0.0    0.0    0.0           -angles for search ellipsoid
```

```
-1                          -number of nearest data to perp. search, or <0 for all
3000000                     -Band width for perpendicular fracture search
3000000                     -Perpendicular distance to search
---------------------------------------------------------------------
***Output Fracture Files***
fracdata.out                -Output GSLIB Fracture Data
fracdata.fab                -Output FracMan Fracture Data
fracdata_optimized.out      -Optimized output GSLIB Fracture Data
fracdata_optimized.fab      -Optimized output FracMan Fracture Data
1                           -fracture set names / numbers
---------------------------------------------------------------------
***Input Fracture Distributions***
spacingdist.out             -Input Actual Fracture Spacing Distribution
angledist.out               -Input Actual Inter-Fracture Angle Distribution
---------------------------------------------------------------------
***Joint Set Parameters***
0                           -Load initial DFN from file? (1=yes, 0=no)
mapdata.out                 -File with Initial DFN
intensity1.out              -Joint Intensity Input File
1 1                         -intensity real. to use, max # of realizations in file
0                           -use locally varying joint orientation (1=yes,0=no)
orientation1.out            -If "Yes", specify locally varying joint orientation file
146.53 11.80                -Mean pole trend and pole trend st. dev.
0     0.001                 -Mean pole plunge and pole plunge st. dev.
65 25 2                     -Fracture Length Mean, St. Dev. and minimum value
2 0.002 0.002               -Fracture Height Mean, St. Dev. and minimum value
0.5 0.1 0.01                -Mean Fracture Aperture, St. Dev. and minimum value
2                           -random z location for fractures? (1=yes,2=no)
2                           -Fracture Intensity Multiplication Factor
---------------------------------------------------------------------
***Other Variables***
3                           -Optimize by intensity and: 1=spacing, 2=angle, 3=all three
0.000035                    -objective function parameter: Intensity constant
850                         -Number of iterations per loop
4                           -Number of iteration loops
2                           -Random Path Option:1=random,2=random without replacement
---------------------------------------------------------------------
***Output Files***
1                           -output objective function progress? (1=yes, 0=no)
iterations.out              -File for iteration objective function progress
results.out                 -File for spacing and inter-angle results
```

The first section of the parameter file specifies a grid in the familiar GSLIB convention. A random number seed is also specified.

DFNSIM uses the same super block search that SGSIM uses. Here the search radii is set to be beyond the extents of the grid so that all data would be considered in the search. For a small problems an exhaustive search can be used, as such, the parameter “number of nearest data to perp. search” is set to -1. The bandwidth for perpendicular fracture search and the perpendicular distance to search are all set to be larger than the grid size so as to not exclude any fractures.

The input distributions of perpendicular distance to the nearest fracture and nearest fracture inter-angle are specified. The input distributions are discussed further in Section 4.

DFNSIM includes the option to load a pre-set initial DFN, which was not used in this case. The joint intensity input is specified as a target intensity of 425 fractures. This version of DFNSIM generates the initial DFN by simulating fracture pole orientation from a normal distribution, specified by a mean and standard deviation. DFNSIM has the ability to incorporate locally varying joint orientations, which can be specified in a file. If locally varying joint orientations are not used, the joint set orientation is then specified directly in the parameter file. The mean and standard deviation for the pole trend, set to 146.53 degrees, standard deviation of 11.80 degrees, are set to match the distribution of fracture poles noted earlier in the paper. The mean and standard deviation for the pole plunge is set to 0 degrees and 0.001 degrees, since the intended fractures are vertical.

Fracture length, height and aperture distributions are simulated from normal distributions by specifying a mean and standard deviation. A minimum possible value is also specified, which truncates the distribution on the lower end in order to prevent extremely small values. DFNSIM simulates in three dimensions. However, this example is a two-dimensional modeling scenario. As a result, it is useful to remove the third dimension by setting each fracture to have the same z-coordinate. Finally, the fracture intensity multiplication factor is specified. The fracture intensity multiplication factor specifies the amount of extra fractures that will be generated. In this example, 425 fractures are required to model the fracture map. A factor of 2 means that 850 fractures will be generated (twice as many as needed).

In the 'other variables' section of the parameter file, the first optimization parameter is specified as 3, indicating that the optimization will use all three components (spacing, inter-angle and intensity). Next, an intensity constant is specified by trial and error that allows activations and deactivations to be accepted, but that keeps the overall fracture intensity within the range of 420 to 430 (where the target intensity is 425). 850 iterations with 4 iteration loops and a random path option of 2 are specified. This means that each of the 850 fractures in the pool will be visited four times (once per loop) in a random order for an activation change. Each of the four loops specifies a new random path.

4. Calculating the Input Distributions

DFNSIM creates a DFN and then optimizes it using an objective function based on three criteria:

1. The distribution of the perpendicular distance to the nearest fracture
2. The nearest fracture inter-angle distribution
3. Fracture intensity

Note that the distribution of the distance to the nearest fracture is used instead of the distribution of fracture spacing (Niven and Deutsch 2010a). The reason for this is that true fracture spacing is much more complicated and time consuming to calculate. Instead the distance from each fracture to the nearest fracture is used as a proxy for fracture spacing. This is discussed further in Niven and Deutsch (2010a).

The program CALCSPACING can be used to calculate the distribution of the perpendicular distance to the nearest fracture, the fracture spacing distribution and the nearest fracture inter-angle distribution. The program CALCSPACING is further detailed in Niven and Deutsch (2010a).

For the digitized fracture data, the distribution of the perpendicular distance to the nearest fracture is shown in Figure 4. The nearest fracture inter-angle distribution is also shown in Figure 4. These two distributions are used as target distributions in DFNSIM. DFNSIM will iterate towards a DFN that respects these target spacing and inter-angle distributions. Ideally the DFN will also respect the number of fracture intersections seen in the observed fracture data since fracture intersections ultimately provide pathways for fluid transport. Two DFNs with the same number of fractures may show very different flow simulation results depending on the fracture connectivity created by intersecting fractures. In this case there are 20 fracture intersections.

5. The Initial DFN

For this example, the initial DFN that was generated to model the fracture map is shown in Figure 5. Upon visual inspection, the initial DFN appears to generate similarly oriented fractures compared to the original map. However, there appear to be many instances of fractures that intersect at various angles as well as many instances of fractures that are located extremely close together.

DFNSIM calculates spacing and inter-angle distributions for the initial DFN prior to calculating the objective function. The distribution of the perpendicular distance to the nearest fracture for the initial DFN is shown in Figure 6. The nearest fracture inter-angle distribution for the initial DFN is also shown in Figure 7**Error! Reference source not found.**. From visual inspection of the two figures, it is easy to see that the initial DFN is a poor match for the fracture map. The initial DFN distances to nearest fractures are smaller than the target distribution. Figure 6 shows that the initial DFN has a wider distribution of nearest fracture inter-angles. This means that, generally speaking, the nearest fractures from the initial DFN are closer together and are oriented more differently than the target distributions. Said another way, the map fractures are more regularly spaced and the fractures are oriented similarly to their nearest neighbors. The initial DFN shows 86 fracture intersections, which is over four times as many intersections as the original fracture map.

Next, DFNSIM iterates by first visiting every fracture location in a pre-determined random order (as was noted above) and switching its activation state. For this example, the target fracture intensity is 425. Here the fracture multiplication factor is set to 2 and the number of iteration loops is set to 4. This means that there are 850 fractures in total, 425 that are activated and 425 that are deactivated initially. Thus, the program iterates 3400 times (850 times per loop over 4 iteration loops).

6. Example Results

Figure 7 shows the optimized DFN after 3400 iterations. Upon visual inspection, the optimized DFN appears to show fractures that are oriented more similarly to their nearest neighbors and more regular fracture spacing, evidenced by fewer bare spots (larger areas without fractures) compared to the initial DFN.

Figure 8 shows the distribution of the perpendicular distance to the nearest fracture for the optimized DFN, which show a reasonable match. Figure 9 shows the distribution of the nearest fracture inter-angle. These two figures show a considerable improvement in the match between the distributions from the optimized DFN and the target distributions from the fracture map. In fact, the match between the target and optimized distributions of nearest fracture inter-angle is very close. The optimized DFN shows 44 fracture intersections, which is over twice as many as the original fracture map, but is half that of the initial DFN.

7. Discussion and Conclusions

Although the success of the optimization depends on the modeling parameters, their choice is relatively straight forward. For small examples such as this, the data search parameters can be ignored by searching all available data. The fracture orientation input parameters were directly measured from the input map data. The iteration parameters were simply set to search all data over four loops. Convergence is discussed in paper 402. However it is noted that four loops of iterations were sufficient to achieve reasonable convergence. The most difficult modeling parameter to select was the fracture intensity constant where trial and error was required to avoid final DFNs with too few fractures. In the initial trials, higher values were used for the intensity constant resulting in no activation changes being accepted. With each trial, the intensity constant was decreased until many changes were accepted, while the final DFN intensity remained within 0.5% of the target value. Using an intensity parameter of 0.000035 resulted in a final intensity of 423 fractures compared to the target of 425.

Future work will include fine tuning the objective function further. It is noted that a better match was achieved for inter- angle than for fracture spacing. There may be a way to adjust the objective function to improve the fracture spacing match without harming the inter-angle match. Other future work will include implementing `DFNSIM` for a large petroleum reservoir example with millions of fractures.

The example presented in this paper shows that it is possible to simulate a DFN with a good match between input distributions of fracture spacing and orientation.

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. A new approach to DFN simulation. Paper 102, 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.

Pana, D.I. and Waters, J.:.G., M. 2001. GIS compilation of structural elements in northern alberta, release 1.0. 2001-01, Alberta Energy and Utilities Board - Alberta Geological Survey, .

Tran, N.H., Rahman, M.K., and Rahman, S.S. 2002. A nested neuro-fractal-stochastic technique for modeling naturally fractured reservoirs. *In* Proceedings SPE Asia Pacific Oil and Gas Conference and Exhibition, pp. 453-464.

Figures

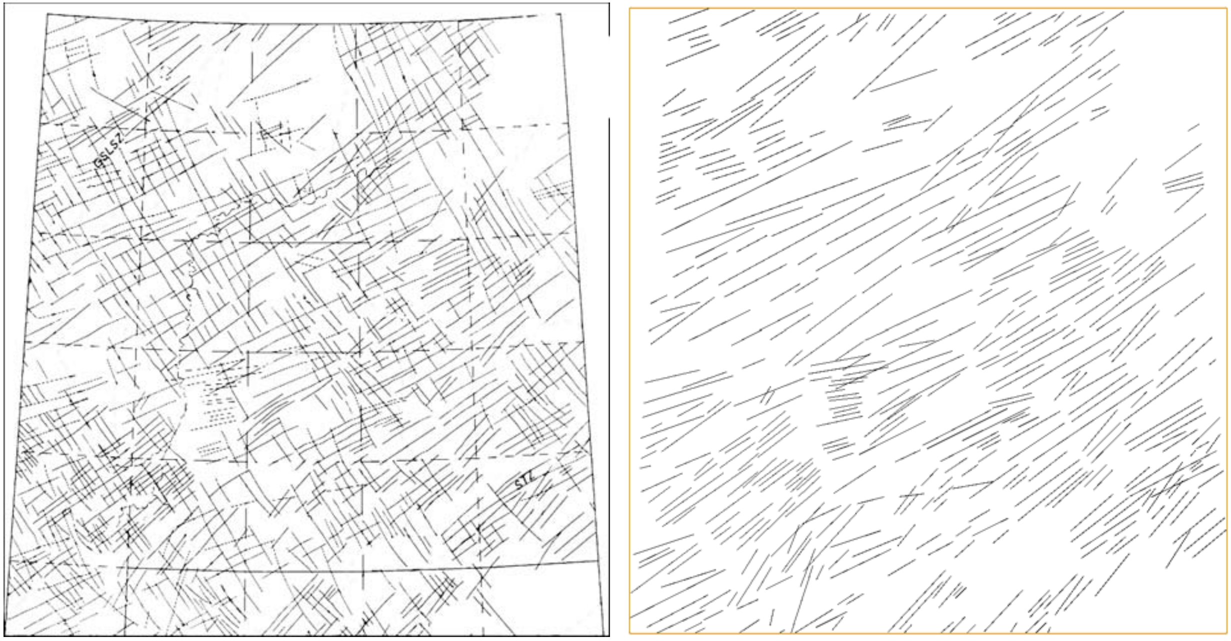

Figure 1: On the left: A map of fractures from Northern Alberta (Pana and Waters 2001). On the right: The digitized SW-NE fracture set.

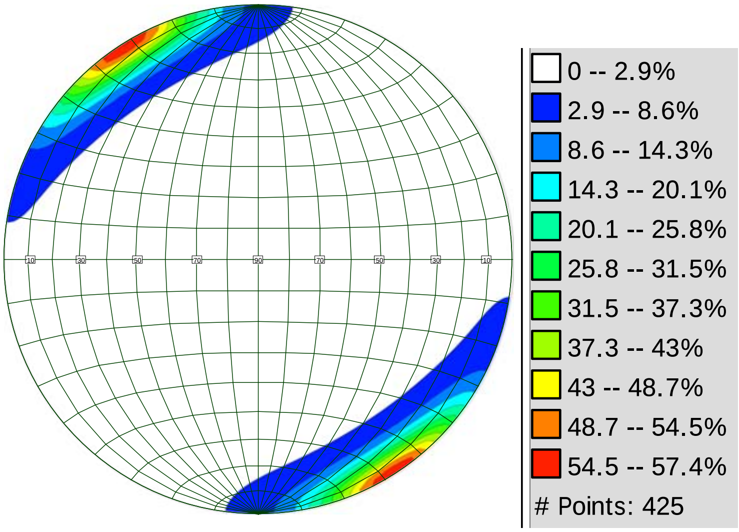

Figure 2: Schmidt equal area stereonet for the fractures shown in Figure 1.

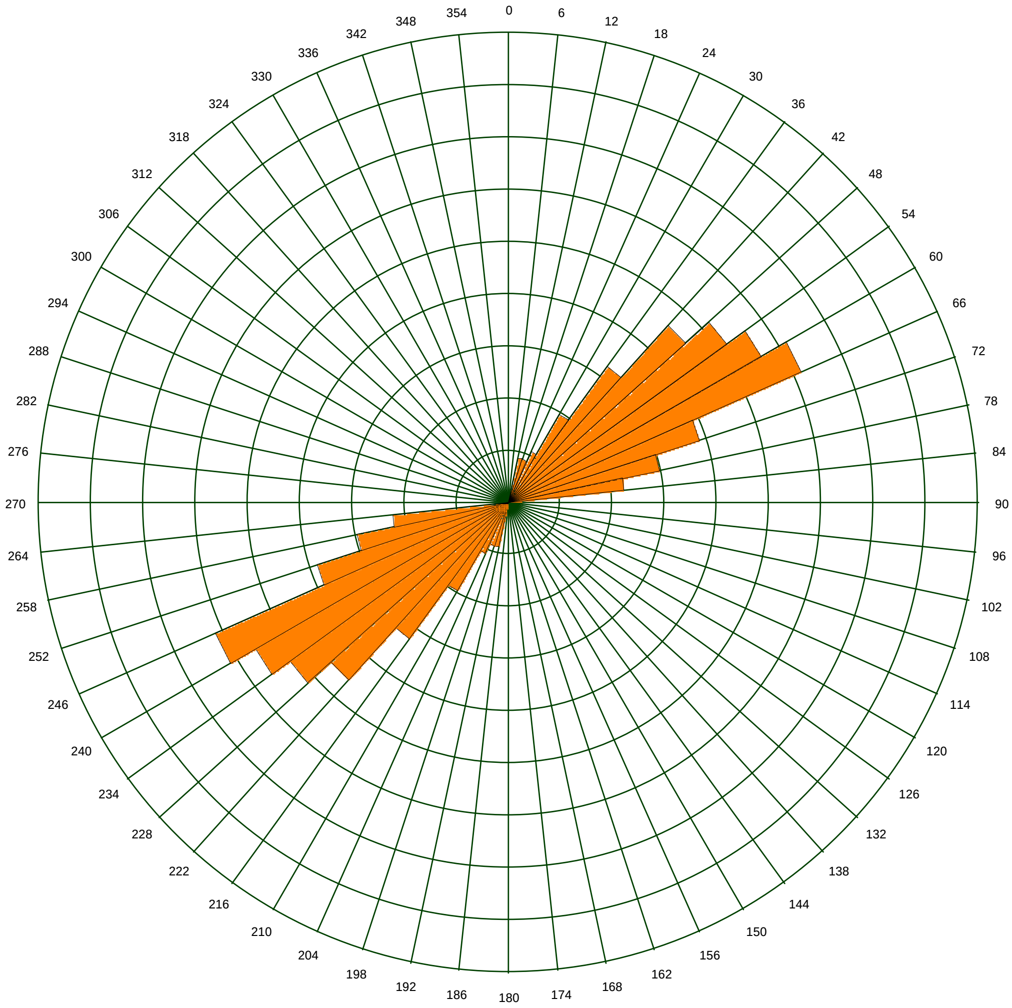

Figure 3: Rose diagram for the fractures shown in Figure 1.

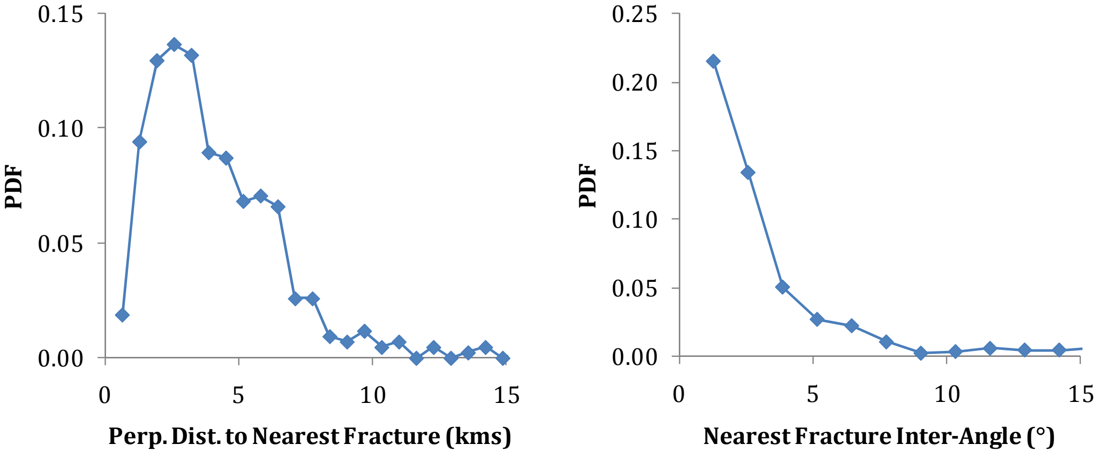

Figure 4: On the left: Distribution of the perpendicular distance to the nearest fracture for the fractures shown in Figure 1. On the right: The nearest fracture inter-angle distribution.

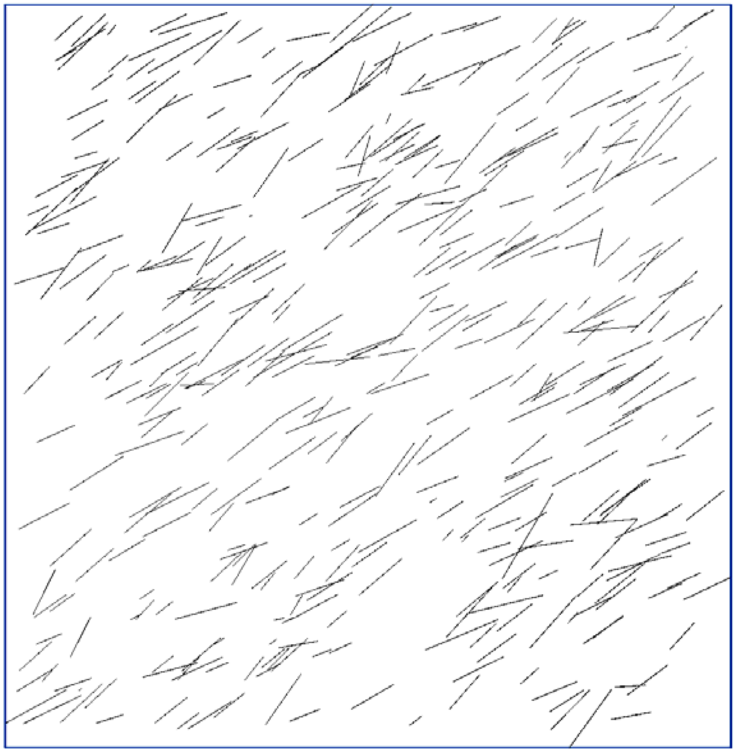

Figure 5: An initial DFN generated to represent the fractures in the fracture map.

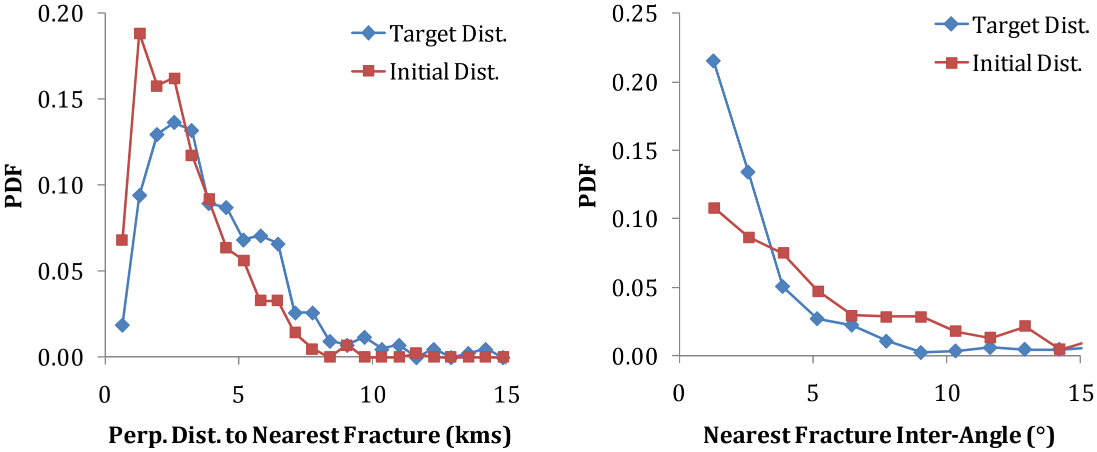

Figure 6: On the left: Distribution of the perpendicular distance to the nearest fracture for the initial DFN. On the right: The nearest fracture inter-angle distribution.

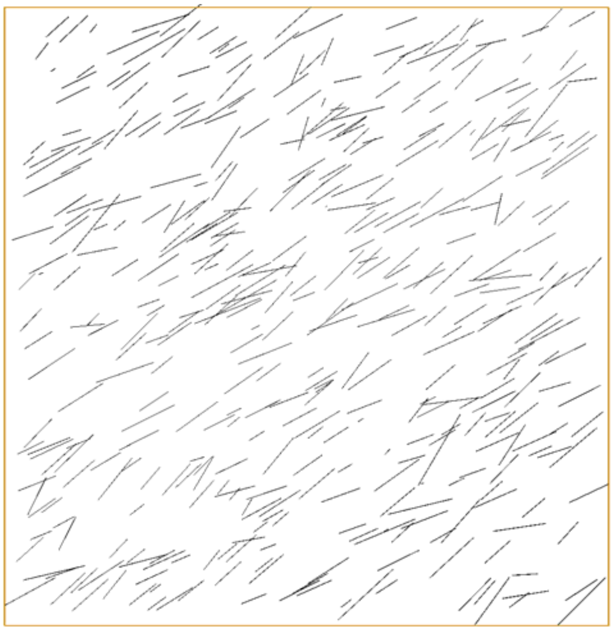

Figure 7: The optimized DFN after 3400 iterations (visiting each of the 850 fractures 4 times).

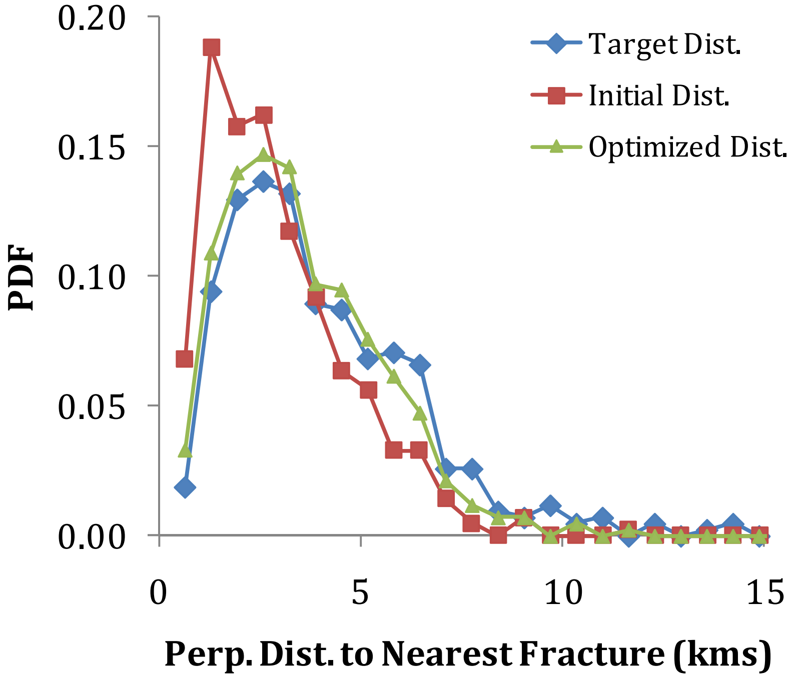

Figure 8: Distribution of the perpendicular distance to the nearest fracture for the optimized DFN.

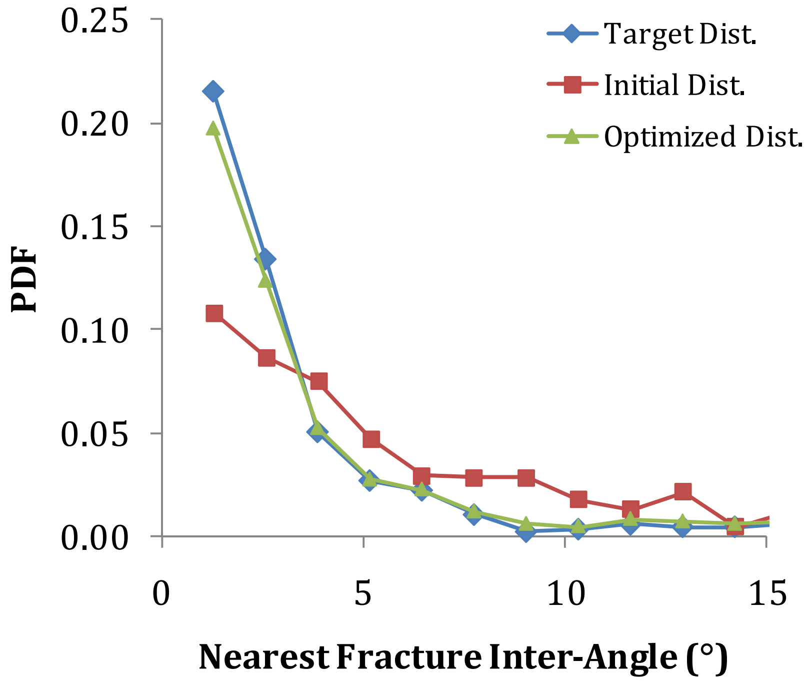

Figure 9: Distribution of the nearest fracture inter-angle for the optimized DFN.