Report

**R-23-02**

June 2024

# Development of new methods to build progressively more deterministic DFN models using data from underground construction and operations

**Tomas Bym**

**Lee Hartley**

SVENSK KÄRNBRÄNSLEHANTERING AB

SWEDISH NUCLEAR FUEL
AND WASTE MANAGEMENT CO

Box 3091, SE-169 03 Solna
Phone +46 8 459 84 00
skb.se

ISSN 1402-3091
**SKB R-23-02**
ID 2049310
June 2024

# Development of new methods to build progressively more deterministic DFN models using data from underground construction and operations

Tomas Bym, WSP Sverige AB

Lee Hartley, WSP UK Ltd

This report concerns a study which was conducted for Svensk Kärnbränslehantering AB (SKB). The conclusions and viewpoints presented in the report are those of the authors. SKB may draw modified conclusions, based on additional literature sources and/or expert opinions.

This report is published on www.skb.se

# Summary

As construction of new underground disposal and accessway tunnels for a repository proceeds there will be requirements to record structural and hydraulic data for both operational and characterisation purposes. Such data can be collated and released periodically as part of a major data freeze and used to update site DFN models, benefiting from additional statistical information or perhaps data from geological domains/structures not previously drilled. The resulting models will be used to support particular regulatory stages. However, as construction and characterisation become more routine, the possibility of more frequent data releases to DFN modelling would enable updating the models conditioned with data from the latest construction areas and thereby increase reliance in the model's ability to predict conditions in the surrounding rock. This opens up the opportunity that DFN models provide an objective real-time aid to planning subsequent rounds of construction and disposal operations. Through routine conditioning of DFN models it is envisaged that stochastic DFN models of the target bedrock would become progressively more deterministic, better reflecting the current understanding and data as it accumulates.

A new method to condition realisations of a fracture network to honour observed traces on surfaces (tunnels and outcrops) and borehole intersects has been implemented in FracMan version 8.0 as a proof of concept. The new implemented method has several advantages over previous algorithms used for conditioning DFN models in FracMan version 7.6. Firstly, it complies to spatial variations in fracture intensity across the model, e.g., where intensity is a function of depth, lithology or proximity to deformation zones. These spatial variations would normally be defined through use of a geocellular model. The conditioned realisations preserve the spatial model defined in the stochastic fracture recipe away from the observation. It also ensures that the full range of possible fractures fitting each trace can be realised over an ensemble of realisations. It achieves this by constructing the specific probability density function of all possible fractures that conform with each individual mapped trace and then samples one of them. The method is fast and capable of conditioning many realisations of large DFN models to many thousands of observed fracture traces.

The method has been verified for several different synthetic configurations of tunnels and boreholes and for different size distributions. It has also been demonstrated to work for genetic type DFN models where the fracture size distribution is an emergent property of a network, grown according to a set of rules for nucleation, termination and arrest, rather than a fixed input.

Further, the same algorithm has also been adapted to the challenge of producing conditioned realisations of fractures representing geologically interpreted deformation zones. Location of these objects which could be both planar and non-planar is commonly interpreted from borehole intercepts but their size is mostly based on geological judgement of the interpreter. The presented algorithm allows the probabilistic assessment of the size effects of these structures.

# Sammanfattning

Allteftersom uppförandet av nya transport- och deponeringstunnlar för ett slutförförvar fortgår kommer det att finnas krav på att kartera strukturgeologiska och hydrauliska data för både drift- och platskarakteriseringsändamål. Dessa data kan samlas in och publiceras regelbundet som del av en datafrys för att sedan ingå, genom det ytterligare underlaget för statistisk information eller kanske data från geologiska domäner/strukturer som inte tidigare undersökts, i arbetet med att uppdatera platsens DFN-modell. Den uppdaterade modellen kommer sedan att användas för att stödja olika steg i licensieringsprocessen. I takt med uppförandet av förvaret blir utbyggnad och karakterisering mer rutinmässig, vilket möjliggör mer frekvent datapublicering så att DFN-modellen kan betingas med data från de senaste byggområdena vilket därmed ökar tilliten till modellens förmåga att förutsäga förhållanden i det omgivande berget. Detta öppnar för möjligheten att DFN-modellen kan vara ett objektivt hjälpmedel för att i realtid kunna planera för kommande steg i utbyggnaden. Genom rutinmässig betingning förutses det att den stokastiska DFN-modellen av berggrunden succesivt blir mer deterministisk, och därmed bättre på att återspegla den nuvarande förståelsen och förutsäga nya data som samlas in.

För att demonstrera konceptet har en ny metod för att betinga realiseringar av spricknätverk mot observerade sprickspår på ytor (tunnlar och hällar) och borrhålsskärningar implementerats i FracMan version 8.0. Den nya implementerade metoden har flera fördelar jämfört med tidigare algoritmer som användes för att betinga DFN-modeller i FracMan version 7.6. För det första kan den rumsliga variationen av sprickintensitet i modellen beaktas, t.ex. om intensiteten är en funktion av djup, litologi eller närhet till deformationszoner. Dessa rumsliga variationer definieras vanligtvis genom användning av en geocellulär sprickmodell. De betingade realiseringarna bevarar således den rumsliga variationen som den definierats i det stokastiska sprickreceptet. Detta säkerställer också att alla möjliga sprickor som kan anpassas till varje spår kan beaktas över en ensemble av realiseringar. Algoritmen uppnår detta genom att skapa en specifik sannolikhetstäthetsfunktionen för alla möjliga sprickor som överensstämmer med varje enskilt karterat spår och därefter sampla utifrån denna fördelning. Metoden är snabb och kapabel att betinga sprickor från stora DFN-modeller till tusentals observerade sprickspår.

För det andra har metoden verifierats mot flera olika syntetiska sprickspårsmodeller från tunnlar och borrhål med olika sprickstorleksfördelningar. Det visas även att algoritmen fungerar för genetiska DFN-modeller, dvs modeller där sprickstorleksfördelningen är en framväxande egenskap hos nätverket, baserad på en uppsättning regler för lokalisering, tillväxt och avstannande istället för en på förhand bestämd fördelning.

Vidare har samma algoritm också anpassats till utmaningen att skapa betingade realiseringar av sprickor som representerar geologiskt tolkade deformationszoner. Placeringen av dessa objekt, som kan vara både plana och icke-plana, tolkas vanligen från kärnkartering, medan deras storlek är mestadels baserad på kartörens geologiska bedömning. Den presenterade algoritmen tillåter en probabilistisk storleksfördelning av dessa strukturer.

# Contents

# 1 Introduction

This document describes the main functionality and theory of a new conditioning algorithm developed in FracMan as part of SKB's DFN Modelling Methodology project. The aim of the new algorithm is to be able to rapidly generate large stochastic DFN models which honour local fracture observations such as tunnel and outcrop tracemap data and fracture borehole logs. It is desirable that conditioned DFN models honour all the observations while maintaining the underlying statistical recipe for each fracture set. In particular, statistics of conditioned realisations over an ensemble should still follow the same size and spatial distribution as for their unconditioned equivalents, assuming the observation data is consistent with those distributions. It is envisaged that as new data is routinely obtained during underground excavations and characterisation, then DFN models can be regularly updated to honour this data, and hence become progressively more deterministic around the construction face.

The proposed method has several advantages over an earlier conditioning method implemented in FracMan for the DFN-R project (Bym et al. 2016, Bym and Hermanson 2018, personal communcation). The new method allows for the statistical recipe to vary in space, e.g., allowing for different fracture domains including those associated with deformation zones. It also ensures that the full range of possible fractures fitting each trace can be realised over an ensemble of realisations.

The conditioning workflow is performed in two steps. The first part of the algorithm constructs a grid of points corresponding to all centres of possible fractures that can meet the constraints implied by a particular trace or set of traces interpreted to belong to the same structure. The second part of the algorithm calculates the probability that the conditioned fracture has a centre at each point in the grid, noting that there can be multiple fractures with different sizes and rotation centred on that point and complying with the observed traces. The probabilities are calculated from the specified fracture size distribution for the fracture set being conditioned, meaning that the size distributions of the fracture sets being conditioned are key inputs to the conditioning process. For Poissonian DFN models, homogeneous or heterogeneous, the size distribution is specified for each fracture. For Grown DFN (G-DFN) models (Libby et al. 2019), the size distribution is an emergent property of the DFN process. However, it can be determined empirically by first growing an unconditioned realisation and deriving a distribution function that fits size of all fractures intersecting a sampling object with traces that need to be conditioned. In next step this distribution function is used for estimating size of conditioned fractures, see Chapter 4.

Compare to previous conditioning algorithm implemented in FracMan 7.6 the proposed approach provides better control when conditioning to multiple observation constraints a rigorous treatment when handling multiple fracture domains, possibility to handle both static and G-DFN fracture sets and it is fast enough to make many realisations of large-scale models. This is motivated by the anticipated need to produce realistic DFN models that include terminations and at the same time condition these fractures to traces mapped in excavated tunnels and surrounding boreholes.

In addition, the Geological modelling methodology (Hermanson och Petersson 2022) has identified a need for new methods to model critical structures that intersect one or more deposition tunnels i.e., so-called CS3 (Critical structures 3, Munier and Mattila 2015) that honour deterministically the geologist's interpretations but take a probabilistic approach to realising the unknown extent of the structure. That is creating realisations of CS3 consistent with both the deterministic rules applied by the geologists and the stochastic distributions and rules inferred in the DFN conceptual model. This method is described in Chapter 5).

Most of the algorithm's basic routines were developed as a part of DFN-R project (Bym and Hermanson 2018) but have been recently updated and/or optimised.

An overview of the algorithm is presented in Figure 1-1. The different items included in this figure are discussed in the remainder of this document, but the summary is presented here to provide context for each component of the algorithm.

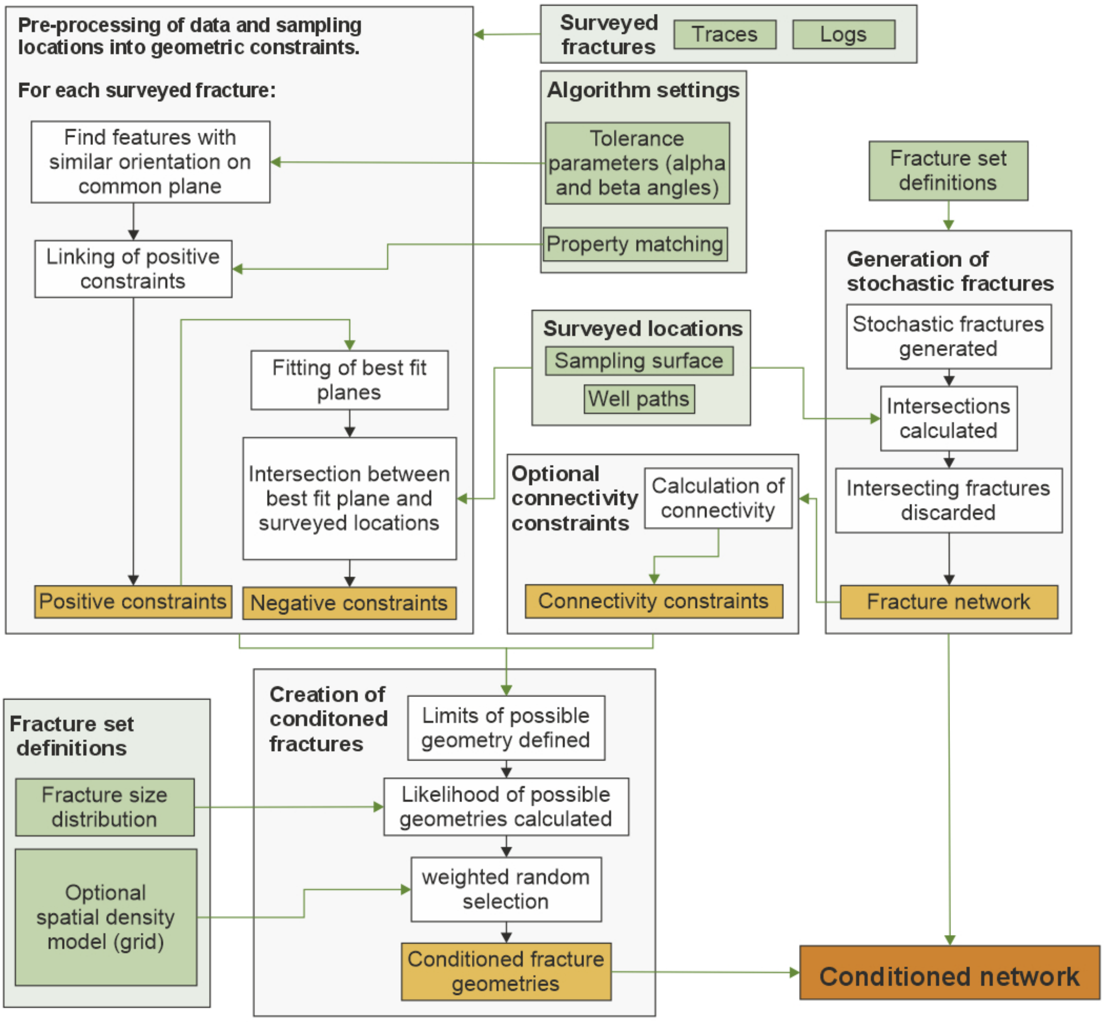

Pre-processing of data and sampling locations into geometric constraints.
For each surveyed fracture:
Find features with similar orientation on common plane
Linking of positive constraints
Fitting of best fit planes
Intersection between best fit plane and surveyed locations
Positive constraints
Negative constraints
Surveyed fractures
Traces
Logs
Algorithm settings
Tolerance parameters (alpha and beta angles)
Property matching
Fracture set definitions
Generation of stochastic fractures
Stochastic fractures generated
Intersections calculated
Intersecting fractures discarded
Fracture network
Surveyed locations
Sampling surface
Well paths
Optional connectivity constraints
Calculation of connectivity
Connectivity constraints
Fracture set definitions
Fracture size distribution
Optional spatial density model (grid)
Creation of conditoned fractures
Limits of possible geometry defined
Likelihood of possible geometries calculated
weighted random selection
Conditioned fracture geometries
Conditioned network

***Figure 1-1.*** *Flow diagram outlining the conditioning algorithm. User inputs are displayed as green boxes. Algorithm inputs are shown as green arrows, outputs are highlighted in orange boxes. Note that for clarity not all aspects of the algorithm are included on this figure.*

# 2 Methodology

Prior to constructing a grid of possible fracture centers the presented algorithm analyses the observed data, searching for possibilities to join multiple observed data by a single fracture and identifying constraints formed by observed objects such as tunnels and boreholes.

It should be noted that FracMan represents fractures as regular n-side polygons rather than circles, and so the algorithm is generalised to work for any n-side polygon, where n is greater than or equal to four, tending toward circles as n increases.

## 2.1 Identification of positive and negative constraints

Identified trace and borehole locations that should be matched by a conditioned fracture are termed positive constraints, and locations where that fracture is absent are termed negative constraints. Positive constraints might be fracture traces in a tunnel wall or fracture intersects in a borehole. Negative constraints are obtained by analysing all possible intersections of a plane fitted to all positive constraints with all sampling objects (tunnel walls, boreholes etc) and comparing the intersection with surveyed data. If the location have been surveyed for structures but there is no mapped structure that would match the plane intersection, this intersection would be considered as negative constraint. Examples include mapped tunnel wall surfaces without a fracture trace or trace having such different properties that the geologist does not consider them part of the same structure, or likewise in a borehole that crosses the fitted plane to positive constraints, but the geologist does not observe that fracture in the core or borehole wall.

Here, an overview of the identification of positive and negative constraints is given, before these aspects of the algorithm are discussed in more detail. In the following text the term sampling objects refers to all boreholes geometrical objects and tunnel/outcrop surfaces. Active trace refers to a traces that is being conditioned by the algorithm in the actual conditioning step.

The first step performed by the algorithm is to analyse the input data and identify all constraints (both positive and negative) for each conditioned fracture. For each positive constraint, the algorithm creates a plane on which the conditioned fractures should lie. The plane is defined either by the geometry of trace vertices (using Orthogonal Linear Regression algorithm to find the best fitting plane to 3D points) or directly by using associated properties representing trend and plunge of the trace. These planes are then used to find other positive constraints that may be part of the same structure and are termed 'best fit plane'.

The best fit planes are also used to identify negative constraints. If a nearby tunnel or borehole does not have a positive constraint proximal to a best fit plane intersection, this object is considered a negative constraint i.e. the conditioned fracture should not create any intersection with any objects identified as negative constraints.

### 2.1.1 Positive Constraints

To identify positive constraints that may be accounted for by the same conditioned fracture, the algorithm can compare the orientation of best fit planes of one or more positive constraints or use a user specified object property.

The comparison of best fit plane orientation uses two parameters termed the Alpha and Beta angles. The Alpha angle defines the size of the search window (wedge shape) within which a potential positive constraint could be found by specifying an angle from the trace plane at trace location (Figure 2-1).

The Beta angle defines the maximum variation in orientation between the trace plane and the positive constraint (Figure 2-2). An angle between the trace plane normal and normal of the potential positive constraint is computed and if it is smaller than the maximum defined Beta angle the positive constraint could be considered for further analysis.

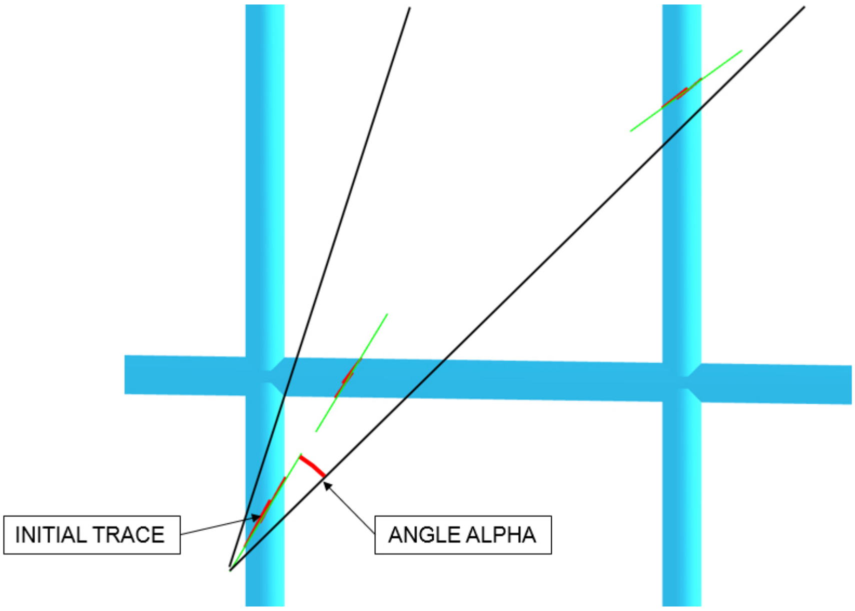


***Figure 2-1.*** *Visualisation of Alpha angle for defining a space of potential positive constraints.*

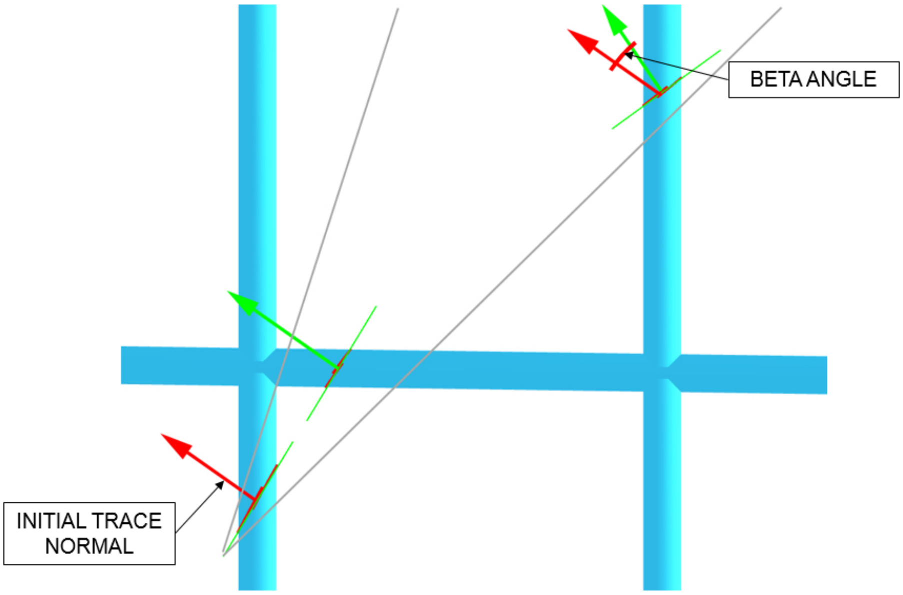


***Figure 2-2.*** *Representation of beta angle defined as the acute angle between normal vectors of the initial trace (red arrows) and potential positive constraint (green arrows).*

Matching of positive constraints using matching values of a user specified property is termed property matching (Figure 2-3). Most commonly this option is used to match traces identified by geologist or by results of flow or tracer tests.

Property matching and the Alpha and Beta angle criteria can be used individually or in combination. If Alpha and Beta angles are not defined default values of 10° and 20° are used. Assuming that fracture surfaces have a self-affine topography (Candela et al. 2009) then the maximum deviation angle of the surface can be estimated to be about 10–15° for very rough fracture and a few degrees for a typical fracture (Stigsson 2016). If none of these parameters are defined, every trace is matched by an individual conditioned fracture.

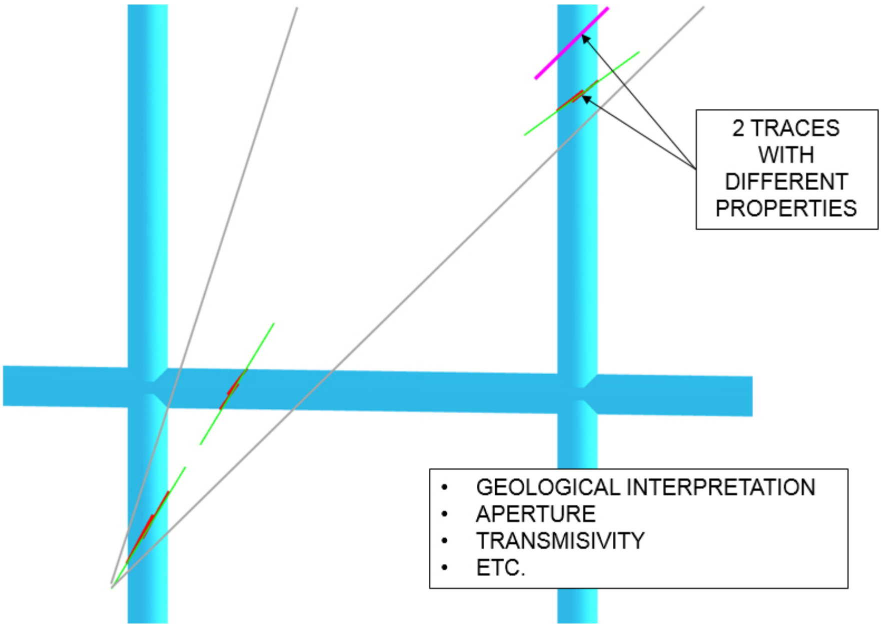

2 TRACES WITH DIFFERENT PROPERTIES
• GEOLOGICAL INTERPRETATION
• APERTURE
• TRANSMISIVITY
• ETC.

***Figure 2-3.*** *Visualisation of the use of property match in identifying positive constraints. Even though both traces fulfil the criteria defined by Alpha and Beta angles only one trace has same specified property value as the initial trace.*

A fracture is indicated to be full perimeter intersection (FPI) fracture if the intersection with the tunnel can be traced around the full perimeter of the tunnel (Munier 2010). Only FPI traces are considered for linking. That means that one non-FPI trace could be connected to multiple FPI traces, but the algorithm does not currently support connecting multiple non-FPI traces.This limitation should not have a strong effect on resulting model as the algorithm would fit each non-FPI trace with a conditioned fracture so the connectivity between the two might still be achieved. In addition, the need to connect two or more non-FPI traces is very rare when working with real data. FPI traces can be linked to multiple FPI traces.

If one or more positive constraints are identified as being able to be connected by a single conditioned fracture the best fit plane is updated to fit all the positive constraints (using Orthogonal Linear regression algorithm).

### 2.1.2 Negative constraints

Negative constraints represent locations with which a conditioned fracture cannot intersect, i.e. it provides an upper bound on the size of the fracture and limits where the centre of a fracture can be. They are identified by creating intersections between all sampling objects and the active trace plane. The algorithm then compares all the positive and negative constraints and removes any potential overlap between the two. It is always the positive constraint that has priority so if both positive and negative constraints are identified at the same location it is the negative constraint that is removed or clipped. This ensures that a negative constraint is not created where a positive constraint was already identified. If a positive constraint is a non-FPI trace, the algorithm is able to create a negative constraint on the rest of the sampling surface.

Note that one of the negative constraints is formed on the same sampling surface. Coloured points on the figure represent probability grid of conditioned fracture centre points (see Section 2.2.1).

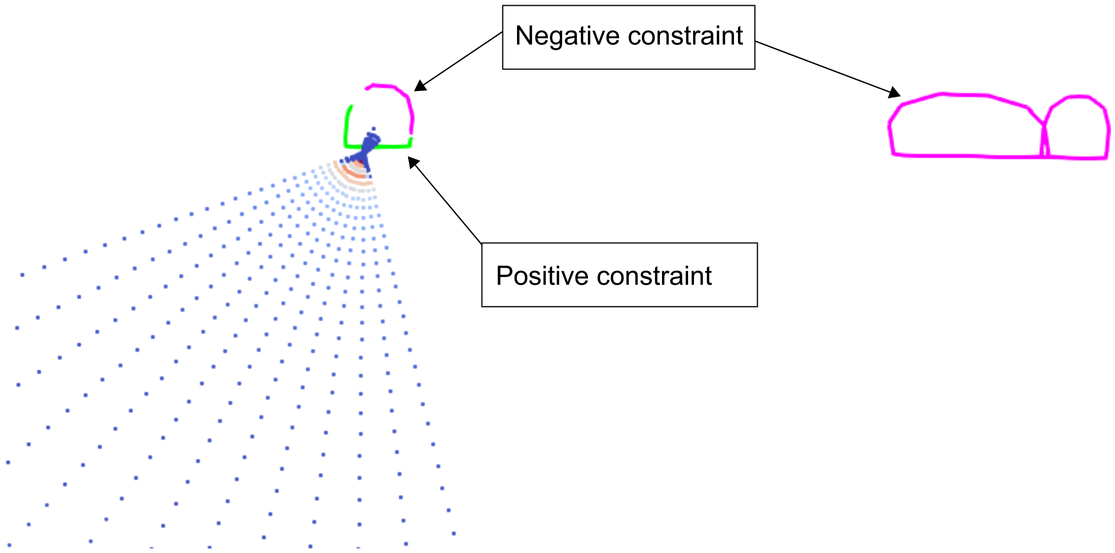


***Figure 2-4.*** *Example of positive (green)for non-FPI trace and three negative (pink) constraints together with the grid points of potential fracture centres coloured by probability weights.*

### 2.1.3 Connectivity constraints

The algorithm allows the user to specify an additional connectivity constraint which ensures that a conditioned fracture and the associated observed traces connect to a specified boundary in every conditioned realisation. This is mostly useful in hydrological modelling when it is desirable to reproduce inflow at specific locations. Using this functionality, the conditioned model honours not only the observed traces and borehole intersections but also the flow/connectivity behaviour (at least geometrically) to user-specified flow boundaries.

Geometrical connectivity is a necessary condition for flow to occur in a conditioned fracture where flow has been observed, but the lack of flow does not necessarily imply a fracture is unconnected, as it may just be a result of low transmissivity in the vicinity of the tunnel, or a hydraulic bottleneck somewhere in the system. Consequently, extra constraints on connectivity are only applied where a flow is observed.

To analyse the connectivity of a fracture network it is necessary that all fractures are generated prior to the conditioning. Once all sets are generated, fractures intersecting sampling objects are removed (the algorithm allows to define a minimal trace length to avoid removing fractures with traces shorter than the user-defined value) and the connectivity of the remaining network to user specified boundaries is analysed.

Figure 2-5 shows an example of how the network connectivity is computed. A flowing boundary represented by the orange surface (Figure 2-5, left) is defined and progressively the connectivity from the boundary is computed. The middle figure shows only fractures that are directly connected to the boundary (connectivity level 1) and fractures connected to these fractures (connectivity level 2). The figure on the right shows the entire connected network coloured by the connectivity level.

Once the network connectivity of the current realisation is analysed, the algorithm starts by identifying positive and negative constraints as described in previous sections. To adhere to the connectivity constraint, the intersection between the generated fracture network and the best fit plane on each positive constraint is calculated, see Figure 2-6. These intersections are then treated by the algorithm as positive constraints, but with the difference that only one intersection per best fit plane needs to be connected by a conditioned fracture.

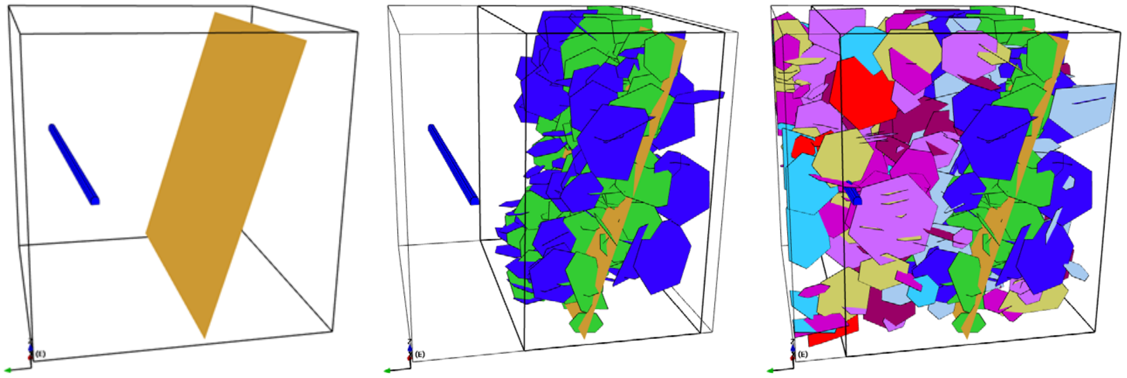


***Figure 2-5.*** *Example simple model with a tunnel and specified flowing boundary (left), fractures intersecting the boundary object with connectivity level 1 and 2 (middle) and all fractures coloured by connectivity level. Fractures with connectivity level 1 and level 2 are displayed in green and blue colours respectively.*

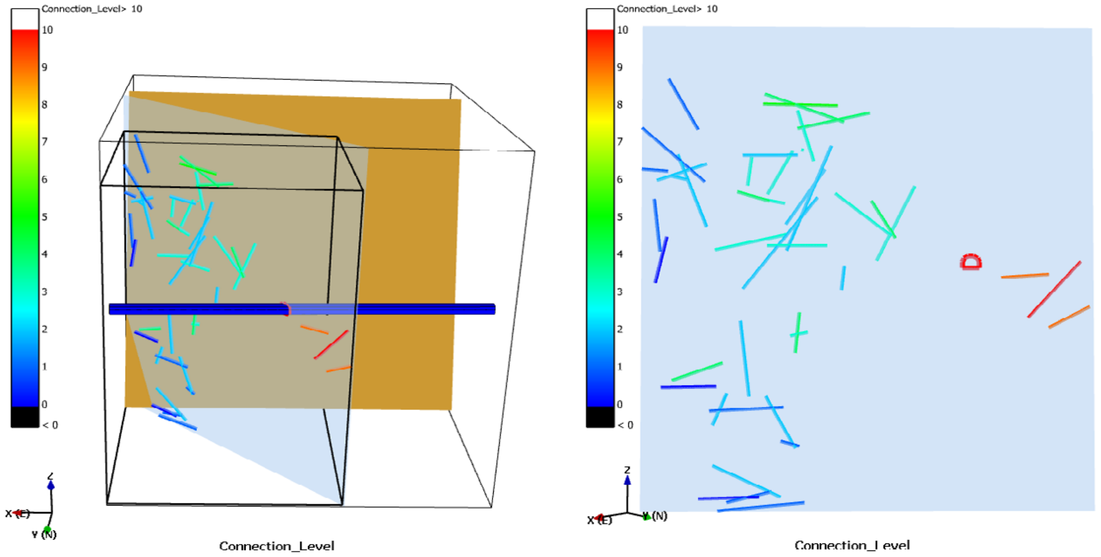


***Figure 2-6.*** *Example of a trace best-fit plane and its intersection with connected fracture traces coloured by connectivity level.*

The algorithm also allows the user to filter fractures based on transmissivity, such that only fractures with transmissivity higher than a user specified value are considered in the connectivity analysis. In addition, for each conditioned trace it is possible to specify the maximum connectivity level of the conditioned fractures. Both features provide better control of the algorithm's behaviour e.g., to filter out tortuous connections with potentially long flow paths along large number of connected fractures or connection through fractures with low transmissivity that might otherwise lead to the conditioned model under-predicting the required transmissivity.

An example of the connectivity can be seen in Figure 2-7. In addition to positive (green) and negative (pink) constraints, traces from connected fractures are included. Note the difference in resulting probability grid points compared to the grid in Figure 2-4 for the same trace. Due to the connectivity constraints, points close to the trace are excluded as the conditioned fracture would not fulfil the connectivity requirements.

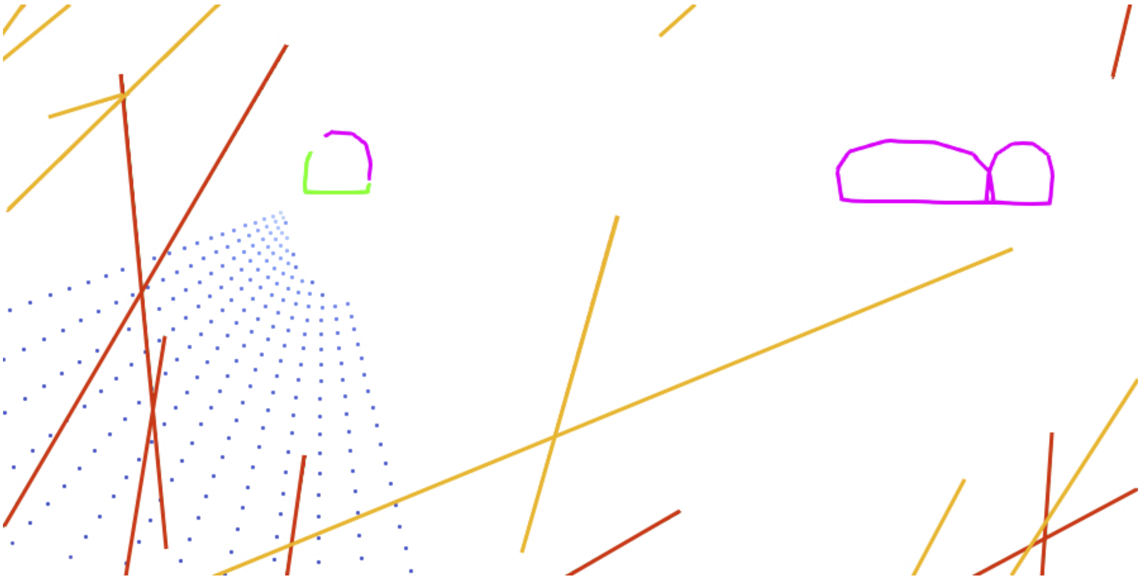


***Figure 2-7.*** *Example of positive (green) and negative (pink) constraints together with the potential fracture centre points. Red and orange lines represent additional constraints (connected fractures intersecting the best-fit plane) derived from connectivity analysis.*

## 2.2 Probabilistic matching constraints

Here, an overview is given of how the positive and negative constraints are used to condition the centres and sizes of fractures probabilistically before these aspects of the algorithm are discussed in more detail.

Using all positive and negative constraints associated with each conditioned fracture, the algorithm creates a grid of points representing potential centres for each conditioned fracture. The centre of each conditioned fracture is sampled from this grid. The probability that any particular point is selected is weighted according to the target size distribution of the fracture set together with the fracture size(s) that matches all positive constraints for the possible centre point.

### 2.2.1 Probability grid

For each conditional fracture, the algorithm starts by translating all positive and negative constraints into a 2D plane to simplify all geometrical calculations. The transformation results in the origin being at the geometric centre of all positive constraints or at the focal point for non-FPI trace (note that there can be only one non-FPI trace in all positive constraints). The grid points are located at regular angles from the origin at distances that increase linearly. This results in a grid as shown on Figure 2-8 where the length of each 'ray' (line of points) is equal to maximum radius defined in the corresponding fracture set definition. Testing showed that using 100 angle increments (3.6° between the grid rays) yields a good compromise between algorithm speed and accuracy.

For non-FPI traces the grid points are restricted to a triangular area centred around the focal point as shown on Figure 2-9. This is caused by the fact that in FracMan all fractures are represented as n-sided polygons. The angle of triangular area depends on number of sides in polygon representing conditioned fracture. If this number approaches infinity, i.e the fracture would be represented as circle, all potential centre points would be aligned on a single line. To increase algorithm accuracy a linear grid of points is used in the area between target trace and the focal point. A linear grid with high resolution proved to be more accurate in representing the possible space of centres of polygons with small radii.

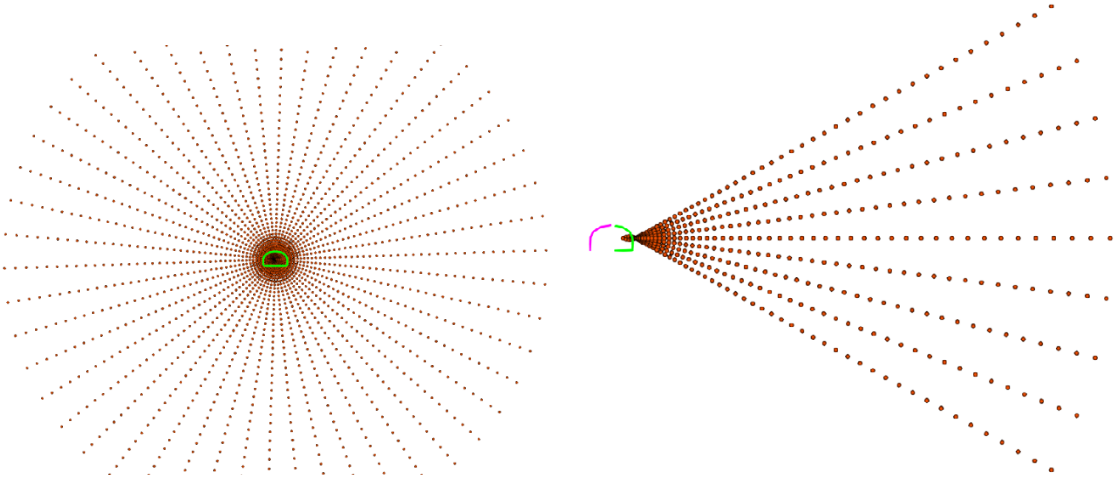


*Figure 2-8. Visualisation of probability density grid of centres for FPI trace (left) and non-FPI trace (right). Green traces are positive constraints, pink are negative.*

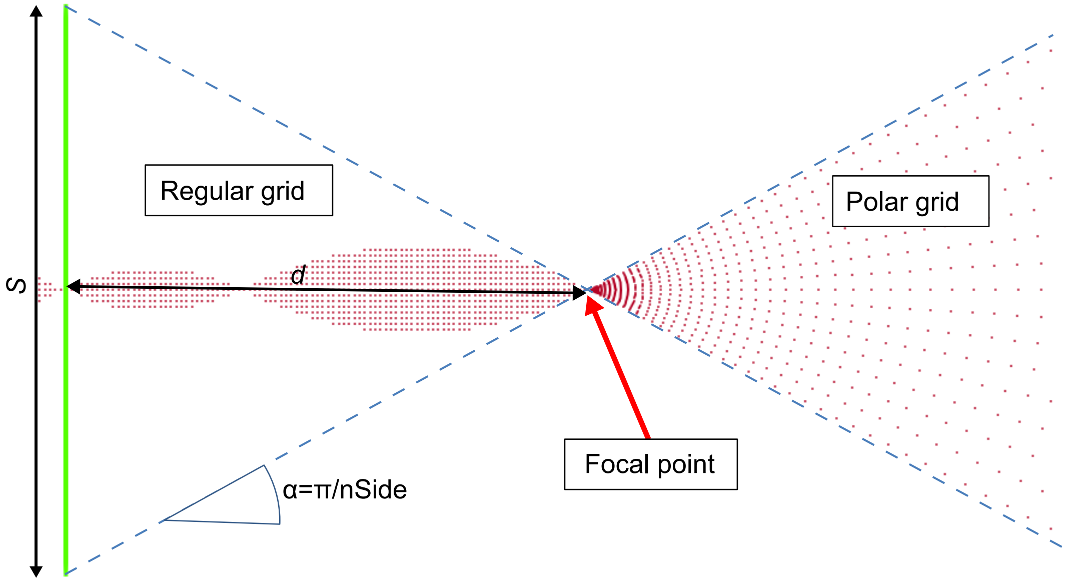


*Figure 2-9. Detail of probability grid for non-FPI trace highlighting the focal point and the two types of grids representing possible centre points.*

The focal point lies on a perpendicular bisector of a segment *S* defined by start and end points of the trace and lies at distance

$d = (dt/2)/\tan(a)$,

where *dt* represents length of the segment *S* and *a* is defined as

$\pi/nSides$

where *nSides* represents number of sides of fracture polygon.

### 2.2.2 Calculating Minimum and Maximum fracture radius

For each grid point the required minimum and maximum radius is computed. For FPI traces, the minimum radius ($R_{c\min}$) is defined as the distance between the point and furthest positive constraint. Similarly, the maximum radius ($R_{c\max}$) for FPI traces is defined as distance between the point and nearest negative constraint. Maximum value of $R_{c\max}$ is defined by upper truncation limit of fracture set size definition. When no negative constraints are identified, the maximum value for $R_{c\max}$ is used. If $R_{c\min}$ is larger than $R_{c\max}$ the point is removed from the grid (effectively assigned 0 probability).

For non-FPI cases, the calculation of fracture radius is more complex due to the representation of fractures as n-sided polygons. For most of the points there are two options for how to match the target trace, either by intersecting the sampling object with odd or even number of polygon sides between the trace endpoints. For most of the points the two options of fitting the fracture polygon are possible so each point contains two possible radii $R_{c1}$ and $R_{c2}$ (Figure 2-10). Details of the calculation are presented in Appendix. For certain points it is not possible to find any polygon that would fit the target trace. Such points are removed from the probability grid. In the last step, the shortest distance to negative constrains is computed and if it is larger than $R_c$ the corresponding point is removed from the grid.

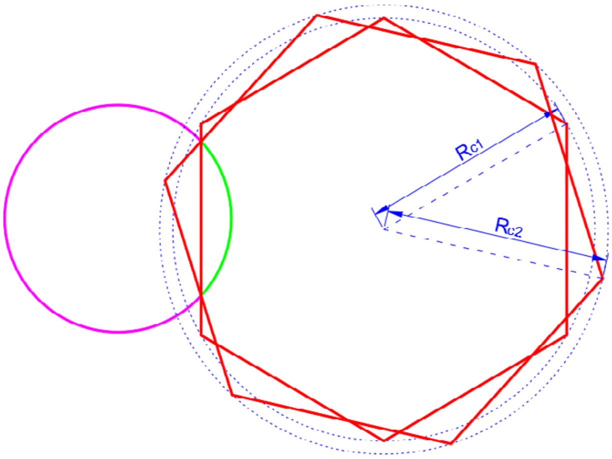


***Figure 2-10.*** *Illustration of two possible ways to fit a trace (in green) with two polygonal fractures with same centre point and radius $R_{c1}$ and $R_{c2}$.*

### 2.2.3 Probability weight calculation

The likelihood of a given fracture centre point being selected is weighted according to the corresponding minimal and maximal radii and the underlying fracture size distribution. The fracture size distribution is defined in the fracture set definition associated to each observed trace.

For FPI traces, the probability of a grid point is calculated as

$$w_i = \left(CDF\left(R_{cmax,i}\right) - CDF\left(R_{cmin,i}\right)\right) A_{cell,i}$$

where CDF represents the Cumulative Distribution Function of the associated size distribution and $A_{cell,i}$, a cell area represented by the centre point as shown in Figure 2-11.

For non-FPI traces the probability is calculated in two steps. First for each point a weight is calculated as

$$w_i = \left(PDF(R_c)\right)A_{cell,i}$$

where PDF represents the Probability Density Function, $R_c$ the corresponding fracture radius and $A_{cell,i}$ the cell area. In a second step we take the minimum $R_{\min}$ and maximum $R_{\max}$ radius out of all grid points and create 100 evenly spaced bins. For each bin the PDF(R) value is calculated, where $R$ is defined as mean value between each bin's upper and lower bound. Next, the sum of all probabilities $w_i$ with radii within each bin's lower and upper bound are calculated and all weights scaled so the sum is equal to each bin's PDF(R) value.

It should be noted that this formulation of probability calculation is only an approximation of rather non-trivial distributions of intersecting fractures.

$$A = \left(\pi r_2^2 - \pi r_1^2\right)\frac{\beta}{2\pi}$$

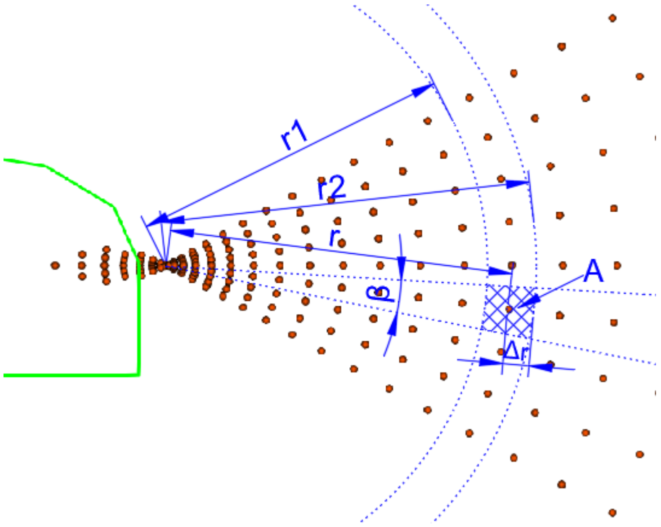


***Figure 2-11.*** *Calculation of cell area in a probability grid.*

If the underlaying fracture definition uses geocellular data with a relative intensity defined as a grid property, it is possible to adjust the probability weight by the grid relative intensity property. For each probability grid point, the probability weight is scaled by the relative intensity read from the grid. This functionality allows conditioning of fractures from geocellular definitions whilst honouring the variable intensity. The example shown in Figure 2-12 demonstrates a grid with a high relative intensity deformation zone and its effect on a conditioning probability grid.

### 2.2.4 Sampling of fracture centre

In the last step, a grid point is randomly sampled from the probability grid using the probability for each point. Rather than choosing one point from a finite set of grid points, all locations within the grid cell, centred around the chosen point, are considered to realise the infinite set of possible centres. The actual fracture centre point is, therefore, randomly selected within the sampled point's cell as shown in Figure 2-11. The fracture polygon is then placed at the new selected centre point and at a specific rotation to match the target trace. For FPI traces, the size of the polygon is sampled from the size distribution with size limits defined by $R_{c\min}$ and $R_{c\max}$ and with random polygon rotation. For a non-FPI trace, the algorithm randomly decides between the Single side and Multiple sides rotations (using the corresponding weights based on $R_{c1}$ and $R_{c2}$) options and computes the required radius and rotation to fit all traces.

A final check of all positive and negative constraints is performed and if all requirements are fulfilled, the conditioned fracture (i.e., centre point, radius, orientation, and rotation) is stored in memory. In the case where some of the requirements fail, a new centre point is sampled, and a new fracture polygon is generated and checked again. If the check fails after 20 iterations the furthest positive constraint is ignored and the algorithm starts again with creating a new probability grid. The positive constraint that is ignored in current step is processed separately so it is still matched by a conditioned fracture. Most common case for ignoring a positive constraint is when a negative constraint is preventing placement of a conditioned fracture fitting all positive constraints. If there is only one positive constraint left and the algorithm is not able to fit a fracture after 20 iterations, the trace is marked as unconditioned.

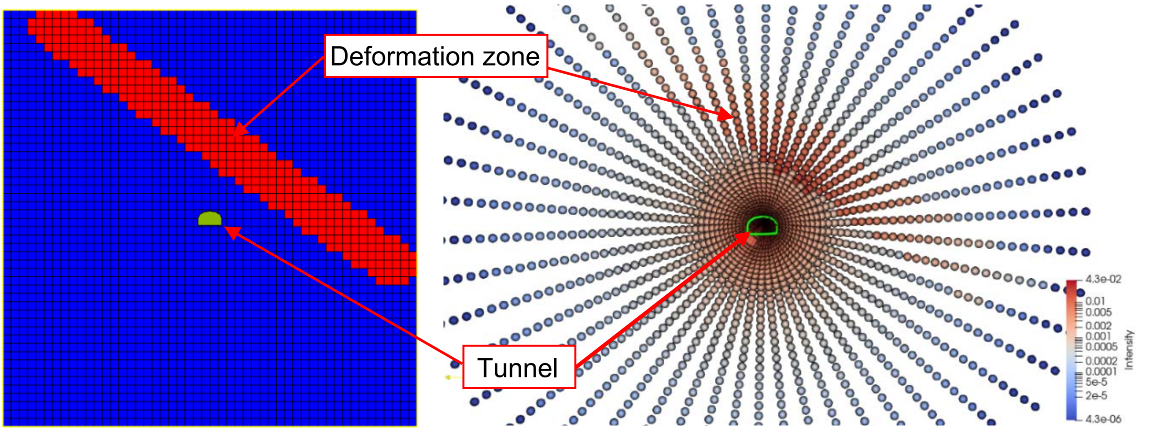


***Figure 2-12.*** *Grid with relative intensity (left) showing a zone with high intensity and probability grid (right) coloured based on the probability weight.*

# 3 Method verification

The algorithm was tested and verified against multiple models in FracMan v. 8.0. Three groups of tests are described here: two that use a synthetic reality which is compared to the conditioned model, and one that uses real fracture data.

The first group of synthetic tests use a simple circular tunnel to test conditioning of fractures with various size distributions. The second group of tests verify the algorithm's ability to condition on traces on multiple tunnels/boreholes using the concept of both positive and negative constraints. The last example tests the algorithm's functionality when real data is used.

## 3.1 Single circular tunnel

Three different size distributions were tested on a simple model including a circular tunnel (see Table 3-1). For details about each distribution and respective parameters refer to FracMan manual (Golder 2022) Fractures were generated in a region of 100 × 100 m with a circular tunnel with radius of 6 m placed at its centre. All fractures have a constant orientation perpendicular to tunnel walls. Depth of the region as well as length of the tunnel was set to 1 m; together with fracture constant orientation this model setup could be considered as pseudo-2D case. For each tested distribution 500 000 fractures were generated resulting in various number of FPI and non-FPI traces which were then used as a conditioning target for the algorithm. CDF plots of fracture size were used to compare statistics of unconditioned and conditioned models to verify that the condition process maintains the fracture size distribution and the spatial model. In addition, heatmap plots of fracture centres were used to visualise the spatial distribution of centres intersecting the tunnel. Results are presented separately for FPI and non-FPI fractures; fractures that don't intersect the tunnel are not considered.

**Table 3-1. Fracture size distribution parameters**.

| Distribution | Min Truncation | Max Truncation | Param 1 | Param 2 |
|---|---|---|---|---|
| Uniform | 5 | 50 | mean = 27.5 m | |
| Log Normal | 5 | 50 | mean = 30 m | deviation = 3 |
| Power-Law | 8 | 50 | $r_{\min}$ = 8 m | $k_r$ = 2.5 |

### 3.1.1 Uniform Distribution

Figure 3-1 shows a comparison of target and conditioned fracture radii. For FPI fractures the conditioning results in slightly larger fractures compared to target. Oppositely for non-FPI traces (Figure 3-1, right) the conditioned fractures tend to be slightly smaller than the original target which is indicated by the conditioned CCN (Complementary Cumulative Number) curve being under the target curve. It should be noted that the presented plots are in arithmetic space (compared to commonly used logarithmic space) so the difference between the two curves is relatively small. The fracture centre heatmap for non-FPI and FPI traces (Figure 3-2 and Figure 3-3) also indicates good match between target and conditioned fractures.

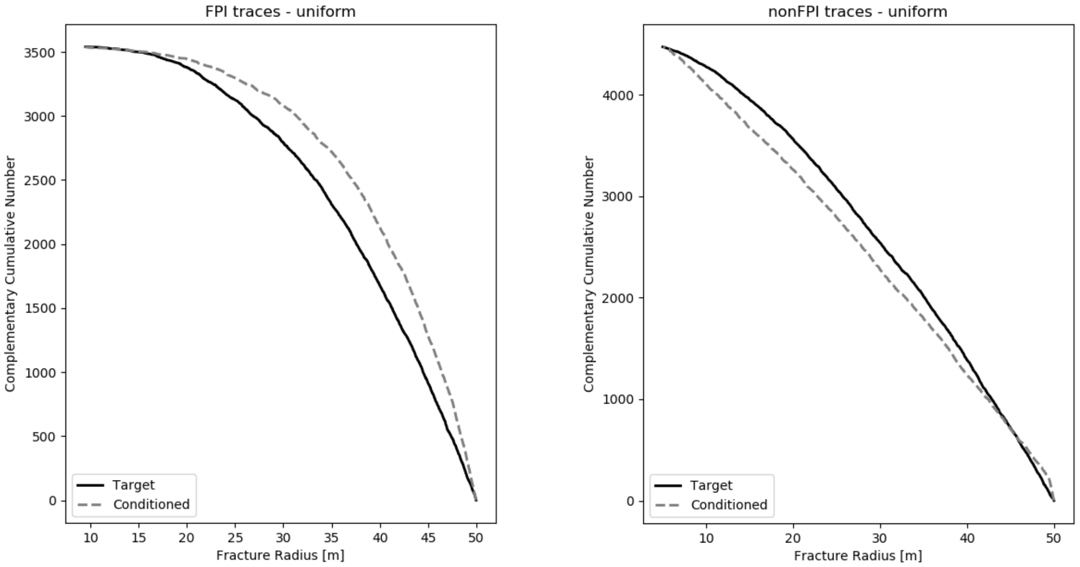


***Figure 3-1.*** *Uniform distribution – comparison of target and conditioned fracture radii for FPI (left) and non-FPI traces (right).*

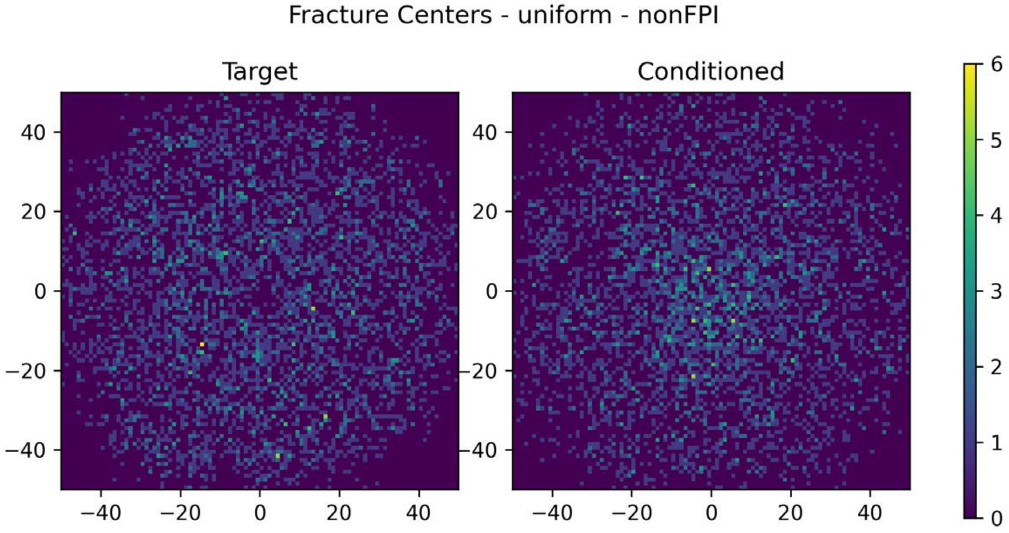


***Figure 3-2.*** *Uniform distribution non-FPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.*

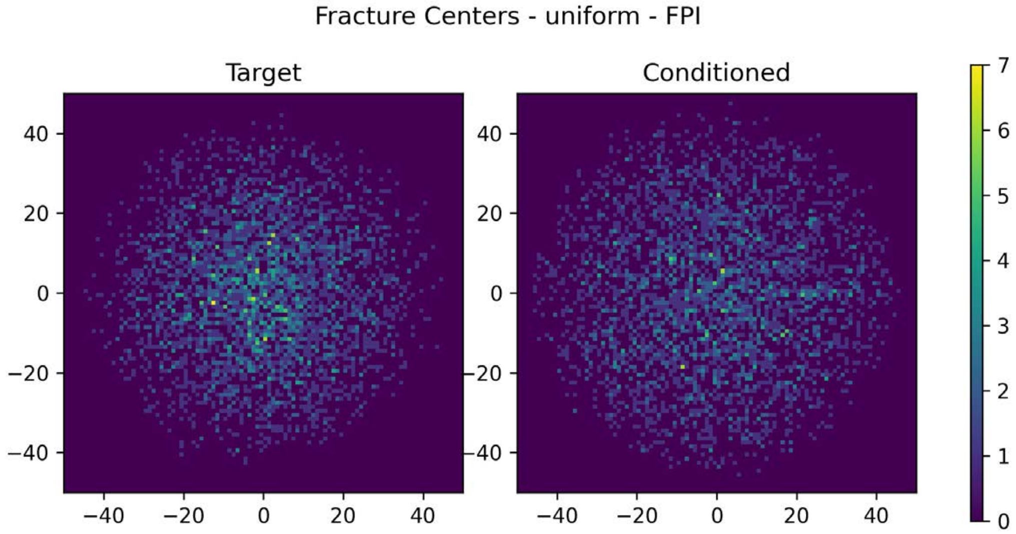


*Figure 3-3.* Uniform distribution FPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.

### 3.1.2 Log-Normal distribution

Results for the Log-Normal distribution (Figure 3-4) show similar pattern for both target and conditioned traces with conditioned FPI fractures being slightly larger than target value. Results show good match between conditioned and target traces both in terms of fracture radius and fracture centres (Non-FPI traces Figure 3-5, FPI traces Figure 3-6).

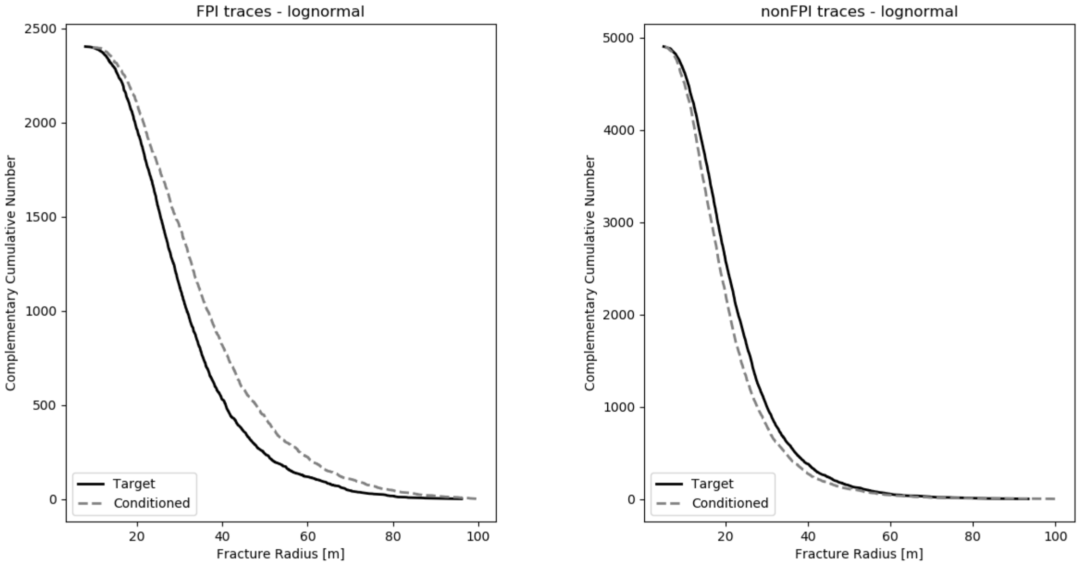


*Figure 3-4.* Log-Normal distribution – comparison of target and conditioned fracture radii for FPI (left) and non-FPI traces (right).

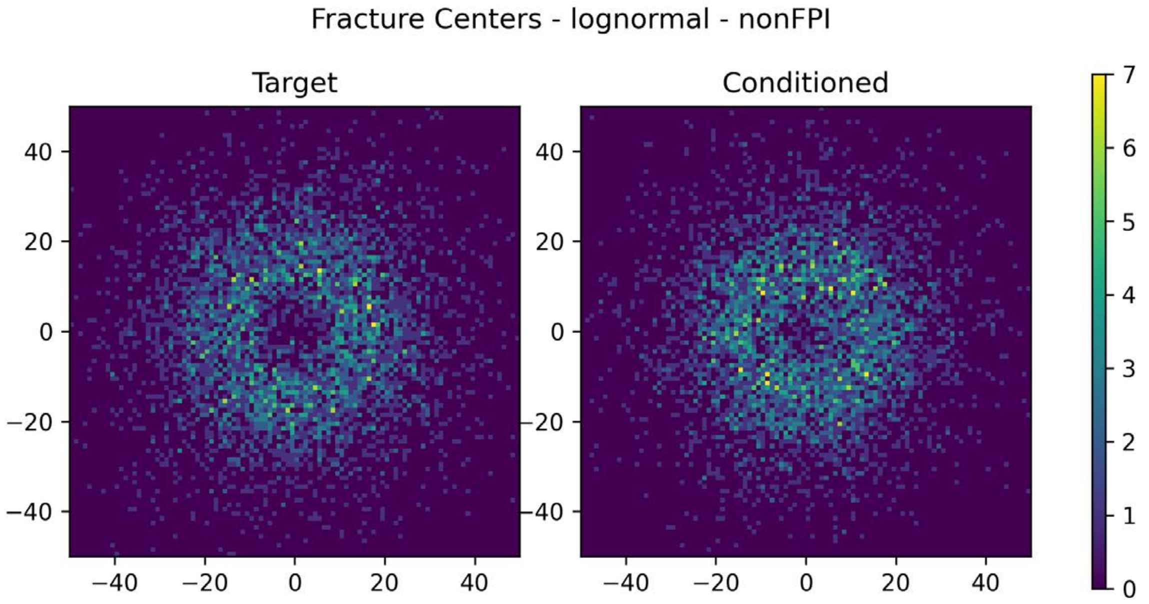


***Figure 3-5.*** *Log-Normal distribution nonFPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.*

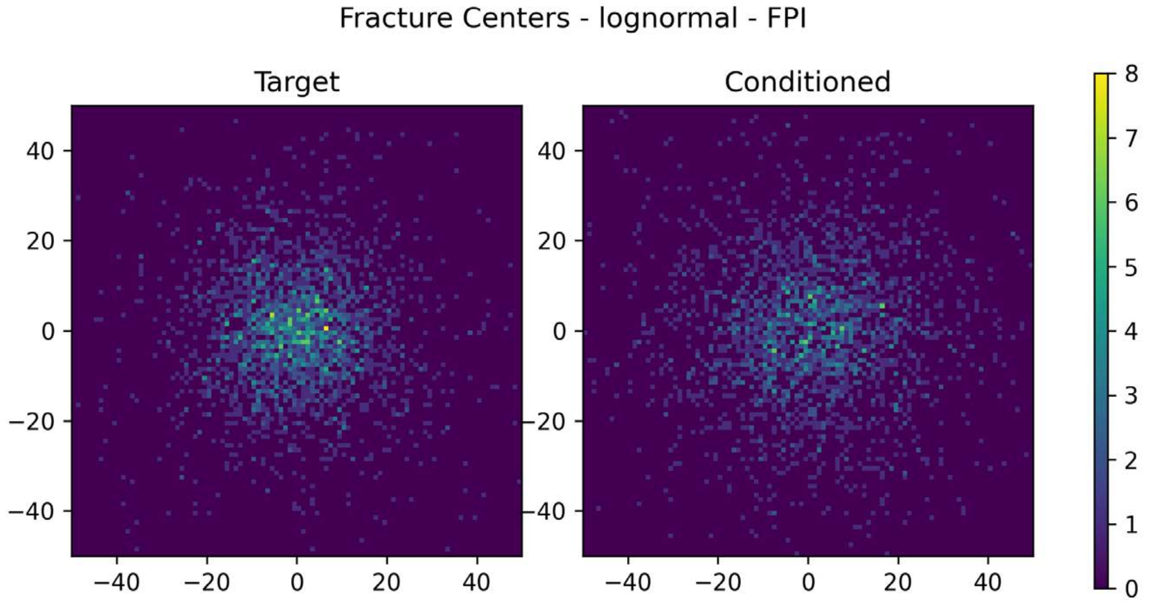


***Figure 3-6.*** *Log-Normal distribution FPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.*

### 3.1.3 Power-law distribution

Results for the Power-Law distribution show a good match between the conditioned and target data in terms of fracture radius distribution for both FPI and non-FPI traces (Figure 3-7). Fracture centres of Non-FPI conditioned fractures (Figure 3-8) show higher concentration in a region around the tunnel center but in general the match is relatively good. This could indicate that the algorithm is predicting slightly higher fracture concentrations around the grid focal point (Section 2.2.1). Opositely for FPI fractures (Figure 3-9) the centre points are more concentrated around the tunnel compared to conditioned fractures but again the match is between the two is relatively good.

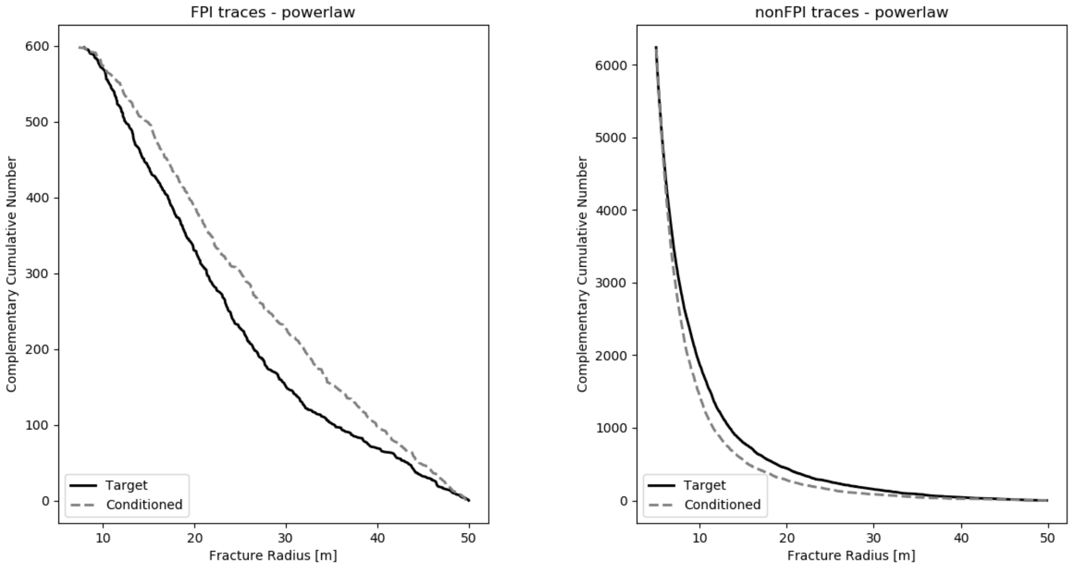


*Figure 3-7. Power-law distribution – comparison of target and conditioned fracture radii for FPI (left) and non-FPI traces (right).*

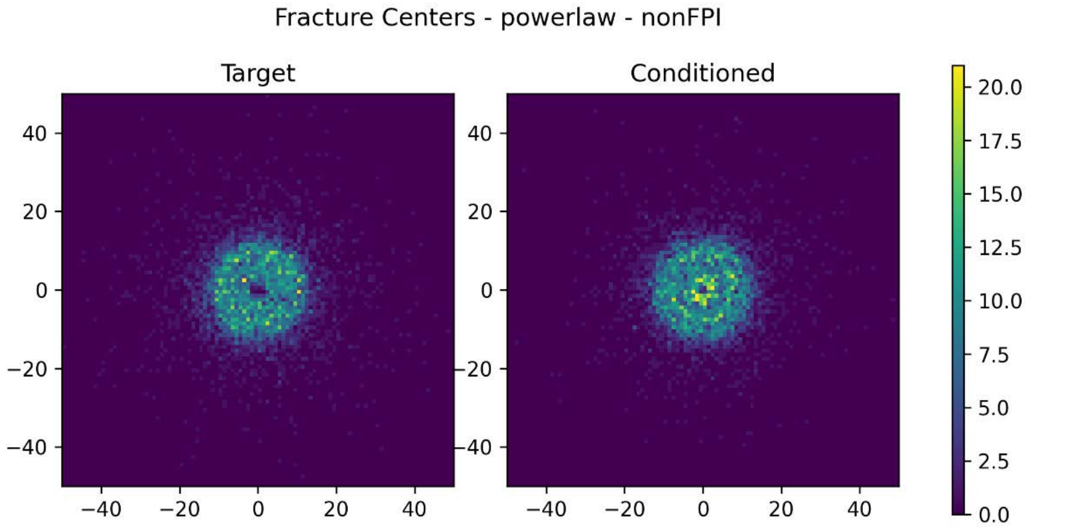


*Figure 3-8. Power-law distribution nonFPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.*

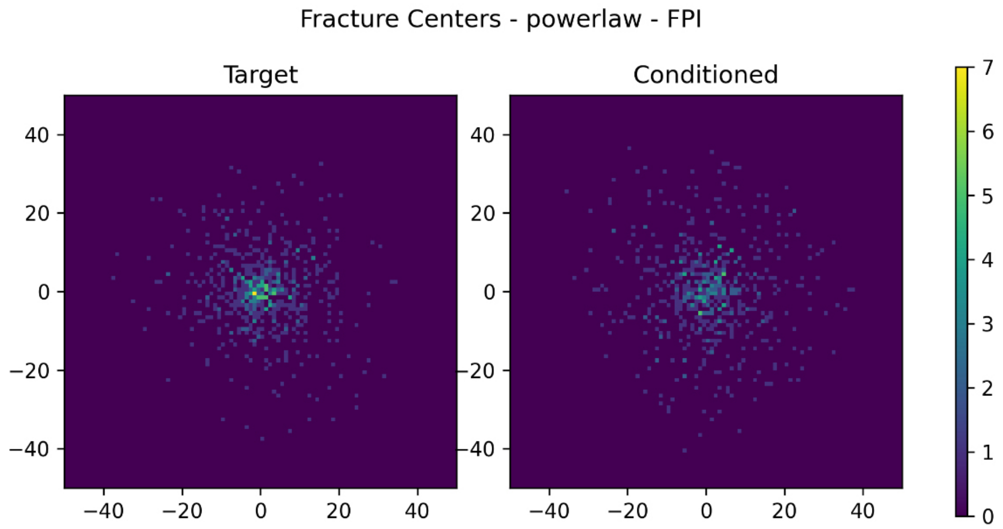


***Figure 3-9.*** *Power-law distribution FPI traces – heatmap plot of intersecting fracture centres showing target (left) and conditioned (right) data. The colour represents the number of fracture centres in the cell.*

### 3.1.4 Summary

Results from the single tunnel test indicate that the algorithm is able to reproduce a similar fracture size distribution for conditioned and unconditioned fractures as well as preserve the spatial variability of fracture centres.

For FPI traces the conditioned fractures tend to be slightly larger than conditioned target radii but the shape of both conditioned and target is preserved for all tested size distribution. Spatial variability of fracture centres indicates slightly higher concentration of target fracture centres around the tunnel compared to the conditioned fractures which tend to be more spread around the tunnel centre.

Fracture size distribution of non-FPI fractures show good match between conditioned and target for all tested distributions. Heatmap plot indicates that conditioned fractures tend to be concentrated closer to the tunnel compared to unconditioned fractures (this is most apparent for Power-law distribution). This could indicate that the algorithm is predicting higher concentration around the focal points of the probability grid.

Overall, the results indicates a good potential for the method to generate conditioned fractures which match target traces and reproduce similar size and spatial variability as unconditioned fractures. The implementation has been generalised to allow any size distributions available in FracMan (Golder 2022) to be used (including those that have not been tested in this exercise). Potential improvement of the method could include the way how centre grid is created and how the probability for each grid point is calculated.

## 3.2 Traceplane, tunnel and borehole

In this test case the algorithm was tested on three different sampling objects. The setup consists of a 200 × 200 × 200 m region box and three sampling objects placed in the centre and defined as a traceplane, circular tunnel with 3 m radius, and a borehole (as a scanline). In a similar manner to the previous test case, fractures with three tested distributions were generated and fracture intersection with the 3 sampling objects were computed. The resulting trace intersections were then used as target for conditioning algorithm.

**Table 3-2. fracture size distribution parameters.**

| Distribution | Min Truncation | Max Truncation | Param 1 | Param 2 |
|---|---|---|---|---|
| Uniform | 1 | 50 | mean = 25.5 m | |
| Log-Normal | 1 | 100 | mean = 10 m | deviation = 5 |
| Power-Law | 1 | 100 | $r_{\min}$ = 1 m | $k_r$ = 2.5 |

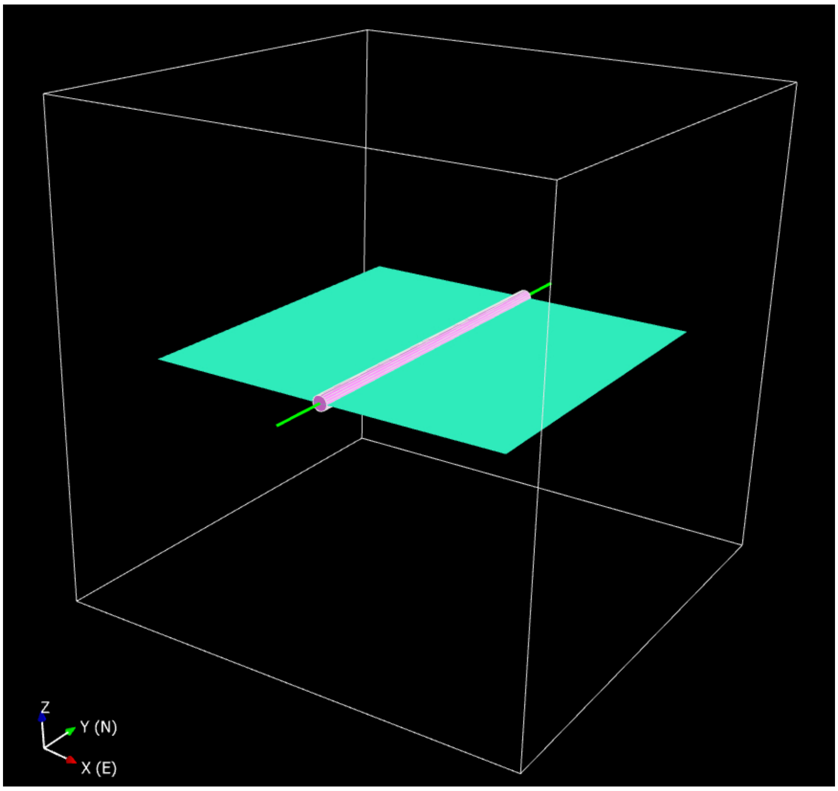


*Figure 3-10. Region box (200 × 200 × 200 m) with traceplane (150 × 150 m), circular tunnel (radius 3 m, length of 150 m) and borehole (length of 200 m).*

### 3.2.1 Results

Results for three tested size distributions are presented in Figure 3-11 through Figure 3-13 both in linear and log-log space. In general, the results show relatively good match between the target and conditioned data especially for smaller fractures. For larger fractures seen on a traceplane, the conditioning algorithm tends to slightly overpredict sizes of conditioned fractures compared to target.

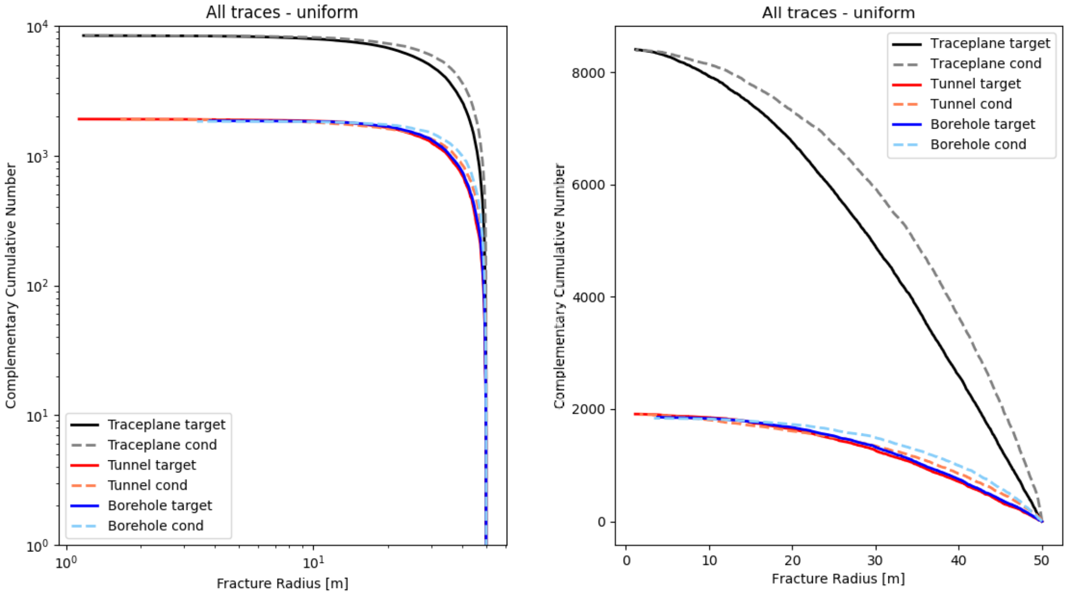


*Figure 3-11. Uniform distribution – Comparison of target and conditioned intersecting fracture radii for a traceplane, tunnel and borehole in log-log (left) and arithmetic (right) space.*

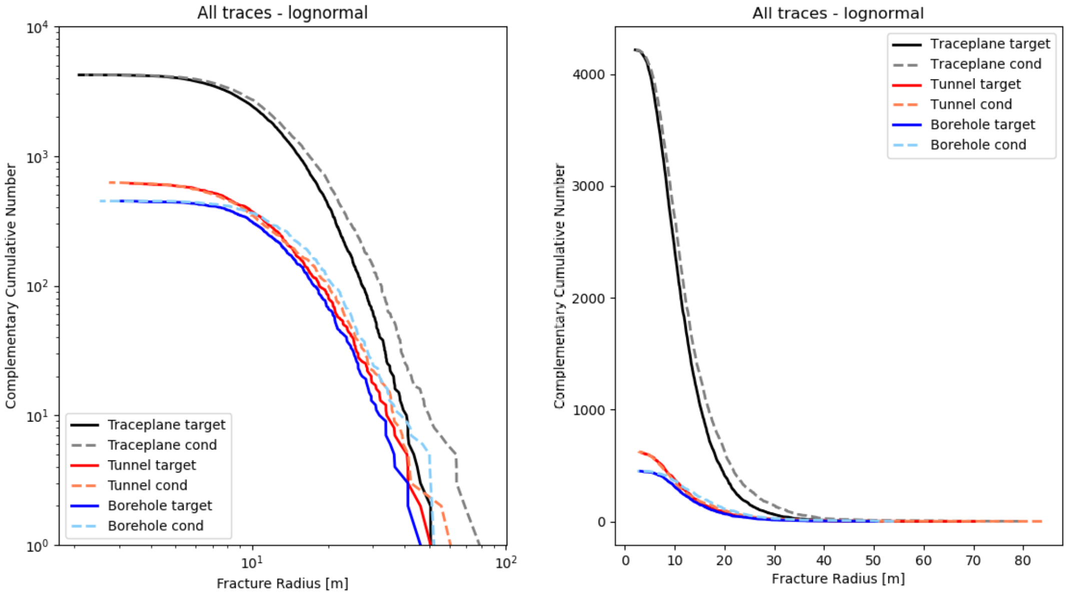


*Figure 3-12. Lognormal distribution – Comparison of target and conditioned intersecting fracture radii for a traceplane, tunnel and borehole in log-log (left) and arithmetic (right) space.*

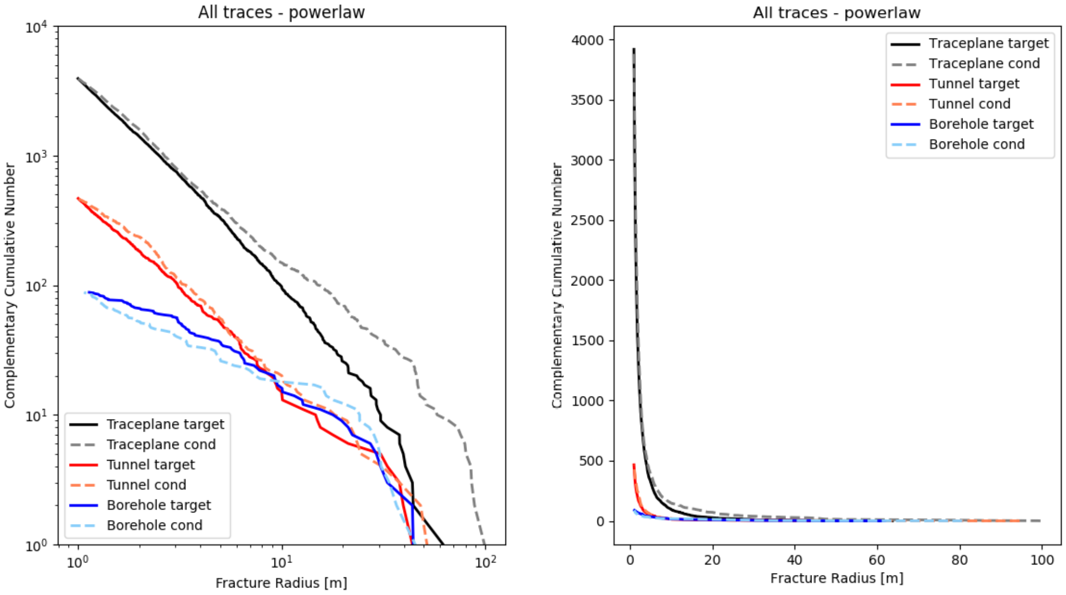


***Figure 3-13.*** *Power-law distribution – Comparison of target and conditioned intersecting fracture radii for a traceplane, tunnel and borehole in log-log (left) and arithmetic (right) space.*

## 3.3 Multiple tunnels and boreholes

To test the algorithm's capability to join multiple positive constraints with a single fracture, a simple model with three tunnels and a single borehole was created. A single set of fractures (with Exponential size distribution with mean value of 30 m and max value of 50 m) was generated and their intersection with the tunnel surface and borehole were computed, see Figure 3-14. The resulting 116 traces and 32 boreholes intersections were used as target data for testing the conditioning algorithm.

Target trace and conditioning results can be seen on Figure 3-14. The algorithm was able to match all traces by using 112 fractures with 28 of the fractures connecting to more than one trace. An example of a conditioned fracture connecting two tunnel traces and one borehole intersection can be seen on Figure 3-15.

Two CCDF (Complementary Cumulative Distribution Function) plots are shown on Figure 3-16. The plot on the left shows trace length distribution of target and conditioned traces. There is almost a perfect match between the two curves. The plot on the right is showing the distribution of fracture radius. It can be observed that conditioned fractures tend to be slightly smaller than the target data, similar to observations as for the uniform distribution in Section 3.1.

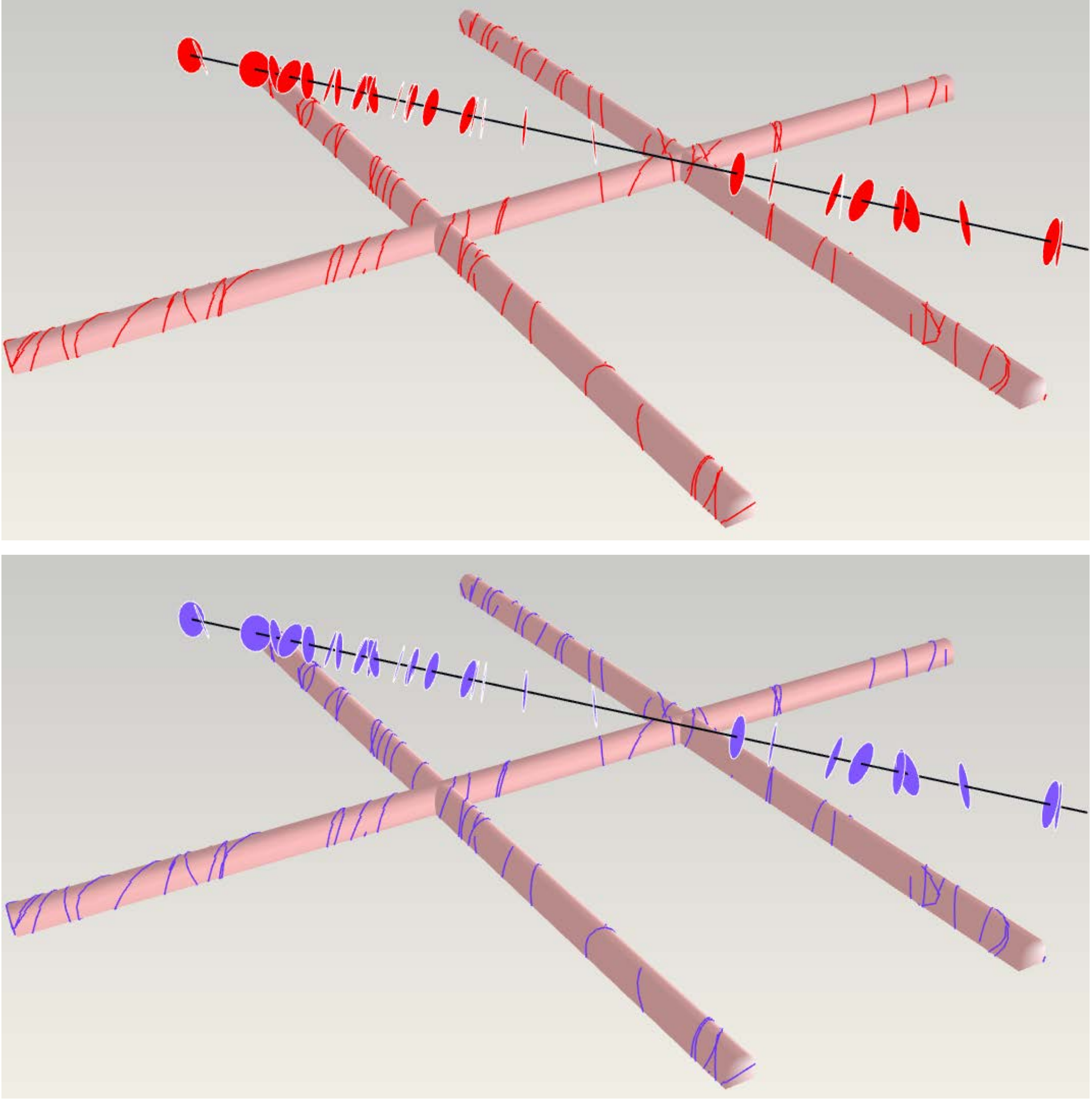


***Figure 3-14.*** *Target traces (a synthetic reality) and borehole intersections (upper figure) and traces resulting from one conditioned realisation on the target data (lower figure).*

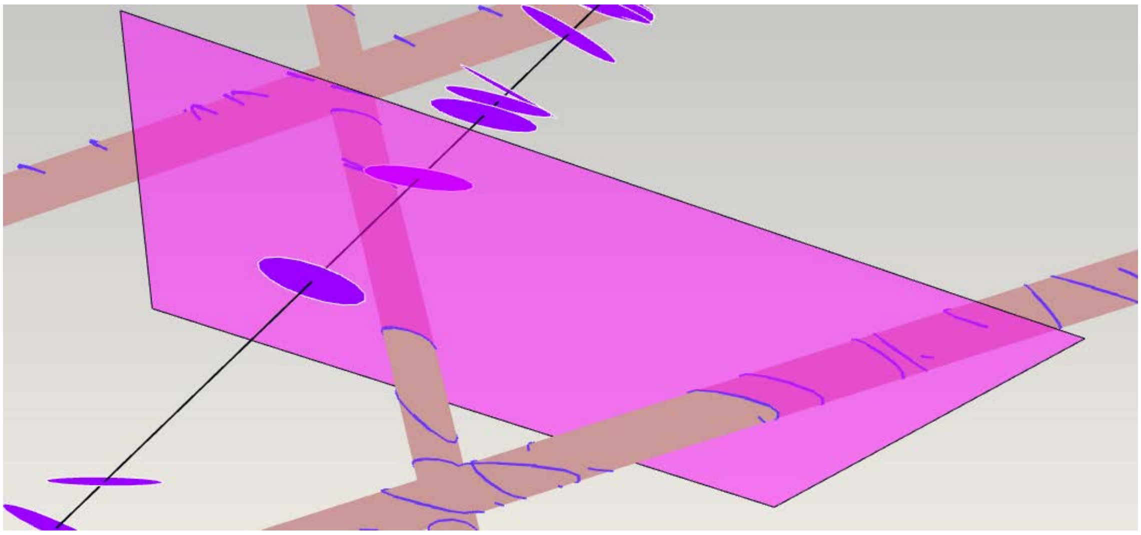


***Figure 3-15.*** *Detail of conditioned fracture connecting two tunnel traces and a borehole intersection.*

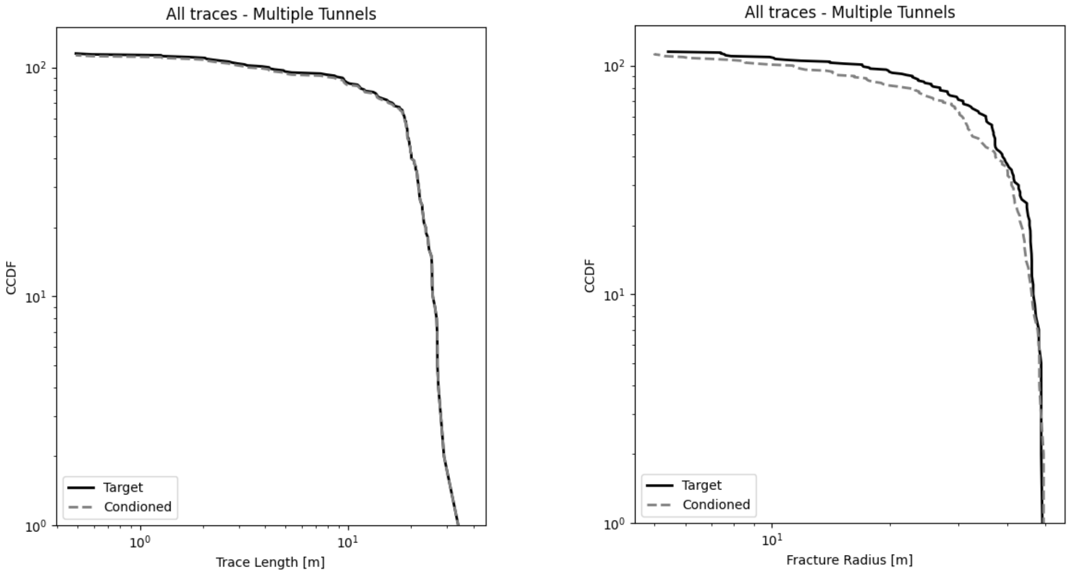


***Figure 3-16.*** *CCDF plot of tunnel trace length (left) and intersecting fracture radius (right).*

## 3.4 Example with real data

To test the algorithm on real data, data from a section of ONKALO access tunnels is used from the Olkiluoto site in Finland. This data comprises fracture traces and a corresponding tunnel surface and a pair of logged boreholes adjacent to the tunnels. A single fracture set (north–south striking) is isolated in the data. This data set was selected as it provided locations where fracture traces may be linked between tunnels, and (potentially) between borehole logs and tunnels. Several pre-processing steps are undertaken using functions within FracMan:

- Traces are relocated to align with the tunnel surface.
- Small trace segments are merged with adjacent segments.
- Traces that nearly formed FPIs are closed i.e., an extra segment is added to ensure the end point of the traces match the start point.
- The roof of the tunnel is removed so that it did not provide a negative constraint for small fractures, as the roof was only surveyed for FPIs.
- All traces under 3 m in length are excluded to reduce the complexity of the problem.
- Some high-intensity sections of borehole log (coincident with deformation zones) were reduced in intensity to reduce the complexity of the problem. These sections could be alternatively represented by a deterministic structures.

The resulting tracemap is presented in Figure 3-17.

The trace orientation is calculated from the fracture trace, and a normal size distribution is used for this illustration. The alpha and beta angles are set to 10 degrees (this value was obtained by trial and error approach to obtain the most geologically plausible conditioned model). Figure 3-18 illustrates one realisation of conditioned fractures produced from these inputs. The traces created at the intersection between the conditioned fractures and the tunnel surface have a size distribution that is a close match to the observed traces (Figure 3-19).

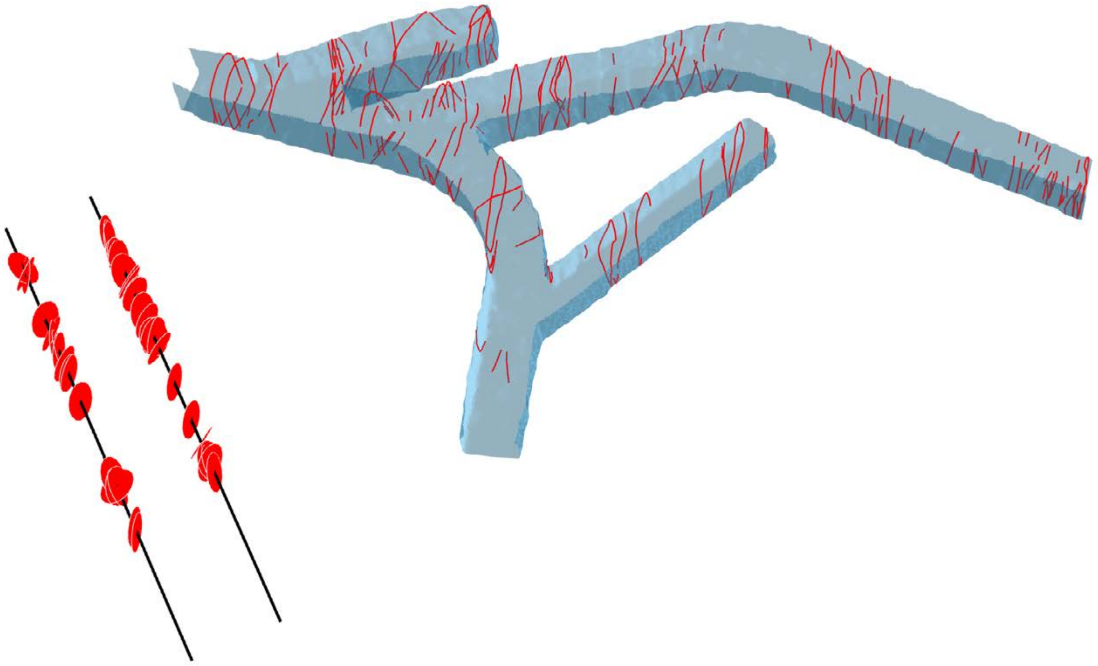


***Figure 3-17.** View of the data used in the conditioning test on real data. Mapped traces both on tunnel surfaces and two nearby boreholes are coloured in red.*

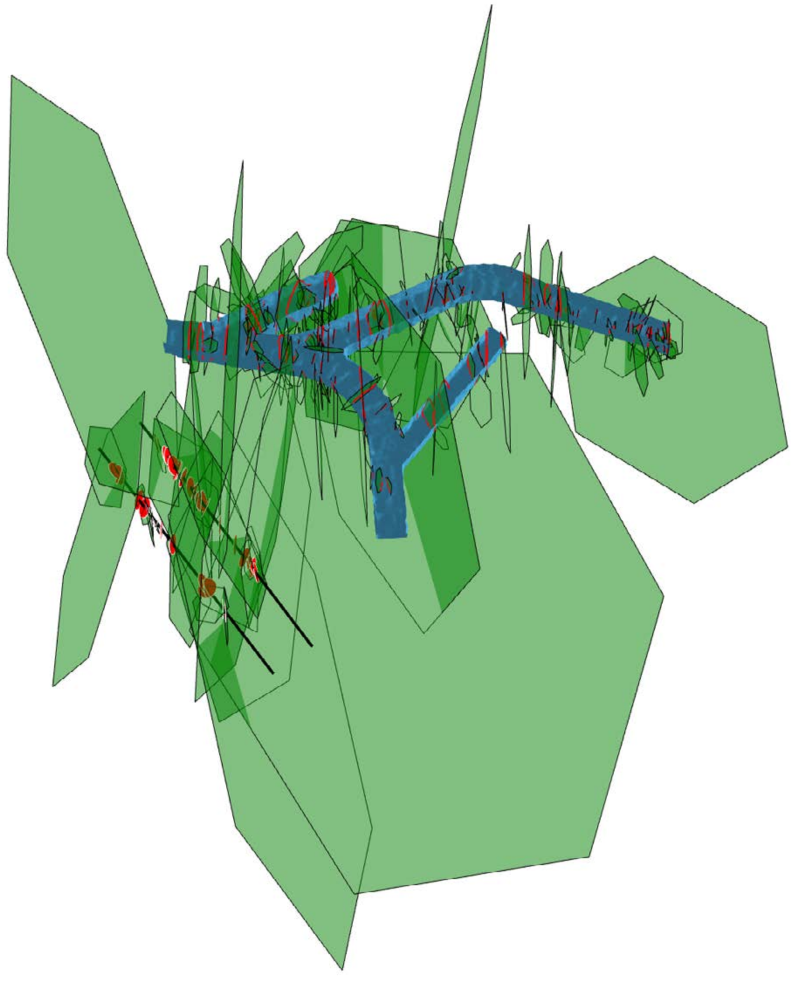


***Figure 3-18.** One realisation of the conditioned fractures produced during this test.*

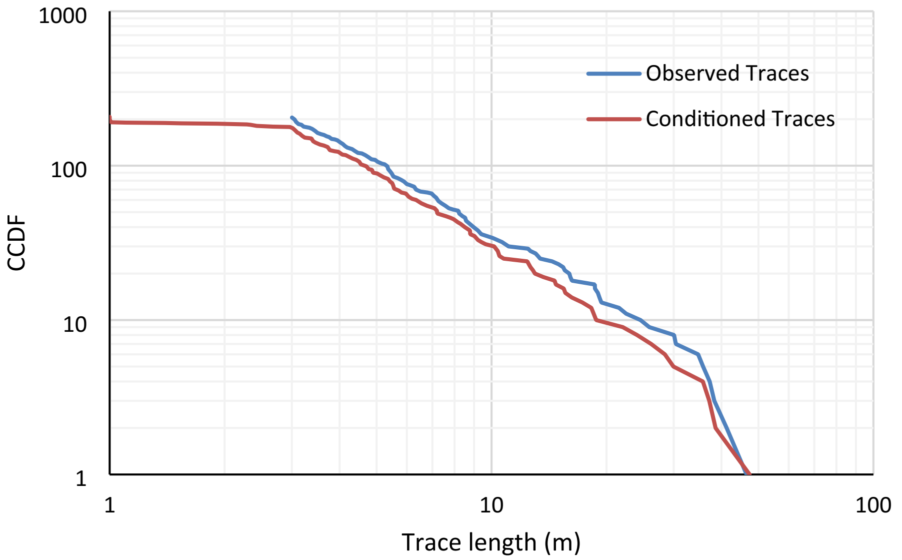


***Figure 3-19.** Complementary cumulative distribution plot comparing the trace length distributions of observed and conditioned fracture traces.*

Several features in the resulting conditioned fractures highlight desirable behaviour:

- Fractures conform to the prescribed constraints, despite complex trace geometries and high-details in the surveyed surfaces (Figure 3-18, Figure 3-20 A).
- Conditioned and observed fracture traces are linked between tunnels (Figure 3-20 B).
- Conditioned and observed fracture traces are linked between boreholes and tunnels (Figure 3-20 C).

However, some features do exist that indicate that conditioning DFNs requires additional quality assurance procedures be applied in maintaining the fracture mapping database:

- Where small gaps exist in the traces, the intervening surface is considered a negative constraint, prohibiting linking of the traces. Development of additional pre-processing techniques for the traces would be required to rectify this (Figure 3-21 A and B).
- Where the traces form complex geometries (Figure 3-21 C). Again, this shows the need to include a step where the raw mapping is repaired to give viable sets of fracture traces for the mapping surfaces used in the conditioned modelling. This pre-processing step was not made here.

It should be noted that the pre-processing step is not part of the conditioning algorithm and should instead be performed by geologists producing the data. While some simple pre-processing methods are included in FracMan or other modelling tools (e.g., merging of nearby trace nodes), the processed data should always be inspected to ensure they realistically represent the mapped features. From the perspective of conditioning, the data should be simple, yet still capture the geometry of the mapped traces. An important requirement is that FPI traces should be formed by a closed polyline.

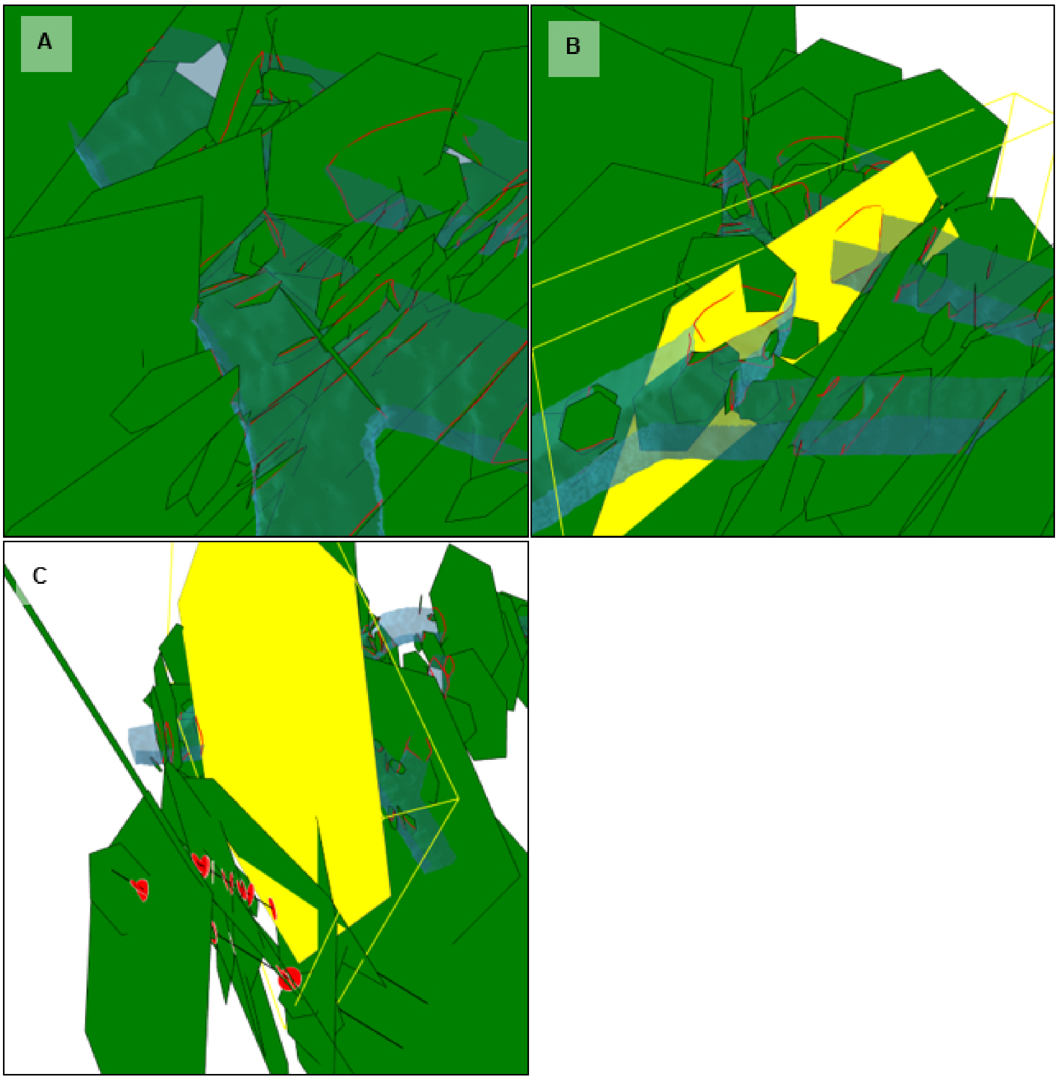


*__Figure 3-20.__ Views of the conditioned fractures produced from the data on different tunnel sections (shown as transparent and blue, with traces in red) highlighting features described in the text.*

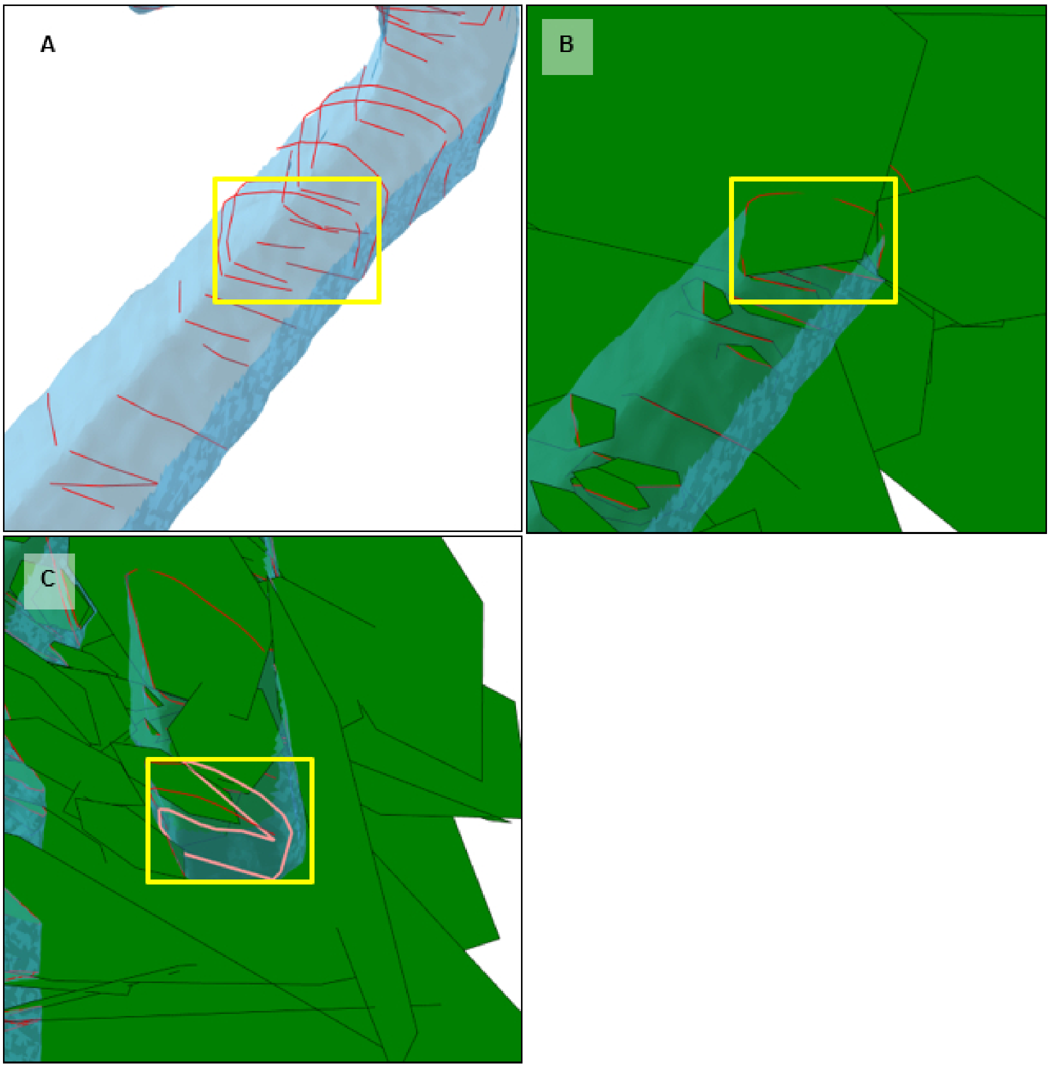


***Figure 3-21.*** *Views of the conditioned fractures highlighting limitation of used tracemap dataset. Figure A and B highlight multiple trace segments which should be joined into a single trace (based on visual inspection). Figure C shows a trace with corrupted geometry.*

# 4 Conditioning of grown DFN models

Grown DFN modelling represents an alternative way of generating fractures in a model which mimics a geological succession of fracture creation. Instead of generating fractures based on underlying fracture definition statistics for statistical models, the Grown DFN (or G-DFN) method allows the seeding and growth of fractures in a transient manner mimicking the deformation history of the bedrock. Each fracture is initiated at a seed point, and it is allowed to grow until its target size (area) is reached or until sufficient fracture terminations against other pre-existing fractures occur. Interaction between growing fractures is governed by simplified mechanical rules which allow modelling of effects such as fracture termination or stress shadowing. For more details on G-DFN refer to (Libby et al. 2019).

In terms of conditioning, G-DFN brings additional complexity as the fracture generation is dynamic where fractures interact with their neighbours, so it is not possible to simply split a model into conditioned and background fractures as it is done for stochastic conditioning (Figure 1-1). In addition, the size of generated fractures is influenced not only by the specified size distribution but also by termination against other fractures, and importantly evolves in time as the bedrock becomes saturated with fractures. With genetic models there is a “dilute domain” (Davy et al. 2013) in space and time where fracture centres are spaced more widely than their size, so they grow undisturbed to meet an *a priori* size distribution, and “dense domain” where intensity is higher or later in the generation process, and size distribution is an emergent property of the termination/arrest processes. As fracture size distribution plays an important role in the presented conditioning algorithm and cannot be directly derived from input data, the workflow for conditioning of G-DFN models needs to be run in two steps. First, to determine the size distribution, and second, to apply it in the conditioning algorithm.

## 4.1 Algorithm workflow

The algorithm for conditioning of G-DFN models shares many modules developed for stochastic conditioning, as described in Chapter 1. The main difference is that, rather than using prescribed statistical distribution for fracture size, one has to be derived specific to the particular fracture set and at specific timesteps. This is achieved by first generating an unconditioned DFN model and calculating bootstrapped size distribution for each fracture set and within a discrete set of time intervals. These bootstrapped size distributions are used in a second pass when the conditioning is performed using the algorithm described in Chapter 1 with the set/time-specific size distributions. In current implementation the generation time for each fracture set (defined as number of timesteps) is split into 10 equal intervals. This was an arbitrary choice here but is used to space out in time the nucleation vs. growth process. The following steps are performed:

1) Fracture size pre-processing
   a. Generate all fracture set definitions. For each set store resulting fracture size distributions in 10 evenly spaced time windows (to approximate the size distribution empirically).
2) For each set in the G-DFN definition
   a. Start fracture set generation. Grow fractures in the model while avoiding intersection with sampling objects (tunnels and boreholes).
   b. In 10 time intervals, sample randomly 10 % of traces for conditioning. For each sampled trace:
      i. Identify positive and negative constraints as described in Section 2.1.
      ii. Calculate the probability grid for the trace (Section 2.2) while using the corresponding fracture size distribution derived from step 1.
      iii. Sample the probability grid and generate conditioned fractures.
         – For FPI traces, set the fracture to minimum required size and allow its growth in successive time steps.
         – For non-FPI traces, generate a regular polygonal fracture with the required size based on the sampled centre (Section 2.2).

### 4.1.1 Fracture Size Pre-processing

Fracture size distribution in a G-DFN model is controlled by the initially prescribed size distribution and termination against other existing fractures in the model. This effect is illustrated in Figure 4-1 where the emergent size distribution of 3 sets is shown. All sets have identical prescribed intensity and size distribution (Power-Law $r_0 = 2$ m, $k_r = 2$). All fracture sets have a termination of 100 % against all pre-existing fractures that are larger than the one which is growing; and sets are generated consequently starting from Set A to Set C. It can be seen that each set results in a different emerging size distribution even though the prescribed input distribution is same for all sets. For Set A, the resulting distribution is close to the prescribed distribution with some limited turnover of the largest fractures. As more fractures are generated, fracture terminations become more dominant, resulting in smaller fractures as seen in Set C, and a curve rather than straight (power-law) distribution arises. For more details on this topic refer to (Libby et al. 2019) or (Selroos et al. 2022).

In addition, fracture size distribution for a particular set is not constant but is dynamically changing as more fractures are introduced to the model. In order to capture this dynamic behaviour, all G-DFN models are first generated without conditioning to provide a resulting bootstrapped fracture size distribution. For each generated set, the creation of the set is split into ten time windows and the fracture size distribution calculated for each. If, for example, a set were generated over 100 timesteps, the size distribution is calculated and stored at timestep 10, 20, 30 and so on.

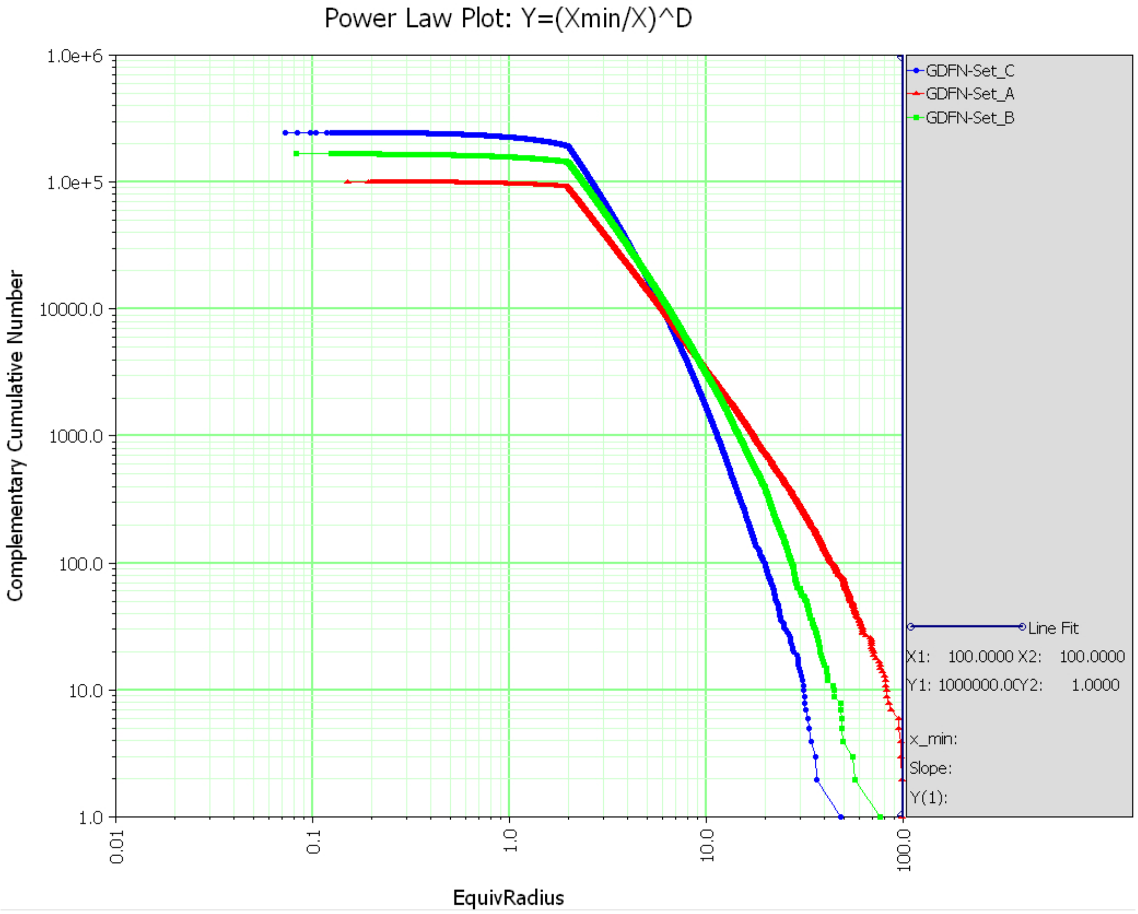


***Figure 4-1.*** *Example illustrating the steepening and turnover of size distribution that occurs for fractures generated at later times due to terminations in progressively more fracture-saturated bedrock. Here, set A, Set B, and then Set C is created. As the fracture number goes up in time, more fractures need to be generated to meet the overall target $P_{32}$.*

### 4.1.2 Fracture set generation

Fractures are generated and grow in a similar way as for standard G-DFN models. Fractures are seeded and grown successively in a prescribed number of timesteps. The only difference between standard and conditioned G-DFN models is that the algorithm prevents fractures from intersecting any sampling objects such as boreholes or tunnel surfaces. If a fracture would intersect any of the sampling objects in the following timestep the algorithm does not allow it to grow any further.

The growth of each set is split into ten equally spaced time windows, in which different bootstrapped size distributions are used. Growth of individual fractures can take place over a higher number of timesteps e.g., some hundreds of steps. The conditioning is split into ten time windows for each set to capture the dynamic behaviour of the fracture size distribution. In each conditioning stage 10 % of target traces is randomly selected. Each trace is processed in a similar manner as described in Section 2.1. A best fitted trace plane is calculated and potential additional traces (positive constraints) that could be formed by a single fracture are identified using the Alpha and Beta angle approach. When the algorithm is trying to identify the potential additional traces in the plane, it considers all input traces and not only the 10 % selected for the current conditioning time window.

Once all negative and positive constraints are defined, a probability grid, as described in Section 2.2, is created. The algorithm assumes all conditioned fractures to be a regular n-sided polygon. For calculating weights for each point on the grid (Subsection 2.2.3) it is necessary to calculate the Cumulative Distribution Function (CDF) and Probability Density Function (PDF) of fracture size distribution corresponding to the particular time window calculated in the pre-processing. The CDF is calculated directly from all sampled data while the PDF is calculated using a kernel density estimation method using a Gaussian basis function.

The conditioning algorithm also considers termination against other fractures when constructing the probability density grid for centres. By using a similar algorithm as described in Section 2.1.3, traces from the intersects of all existing fractures with the current trace plane are obtained to form extra constraints. For each point on a probability grid and considering required fracture radius, intersections with existing fractures (traces on current trace plane) are calculated. For each intersection, a random value is sampled and compared against the specified termination probability in the G-DFN definition. If the result is true, i.e., the fracture should terminate, the point is removed from the probability grid for centres.

This is illustrated in an example shown in Figure 4-2. In this example, three potential fractures are tested for termination against existing fractures. If a termination probability is set to 100 % (fractures terminate against all existing fractures), it is only the smallest green fracture that would be suitable for conditioning since the other two fractures intersect at least one pre-existing fracture. If a termination probability is set to 50 %, say, then the probability of one of the three fractures to be accepted for conditioning is as follows. For the smallest green fracture, the probability is 100 % since it is not affected by any intersection, for the larger blue fracture the probability is 50 % with one intersection, and for the largest red fracture the probability is 25 % with two intersections. Note that the red fracture has actually three intersections but the intersection marked by black dashed line is ignored because of the G_DFN rule that a fracture can terminate only on a larger fracture.

The algorithm samples a point from the probability density grid, computes intersections with existing fractures and samples the termination probability. If a point fails, i.e., the fracture will terminate against at least one existing fracture, the algorithm samples a new point. If after 200 tries the algorithm is unsuccessful in finding a suitable point, the termination restriction is ignored for the current trace. If successful, the point becomes the centre of a grown fracture. For a FPI trace, the initial size is set to the smallest that makes the required intersect, then grows beyond that, while for non-FPI trace, the fracture's size is fixed by the end points of the trace.

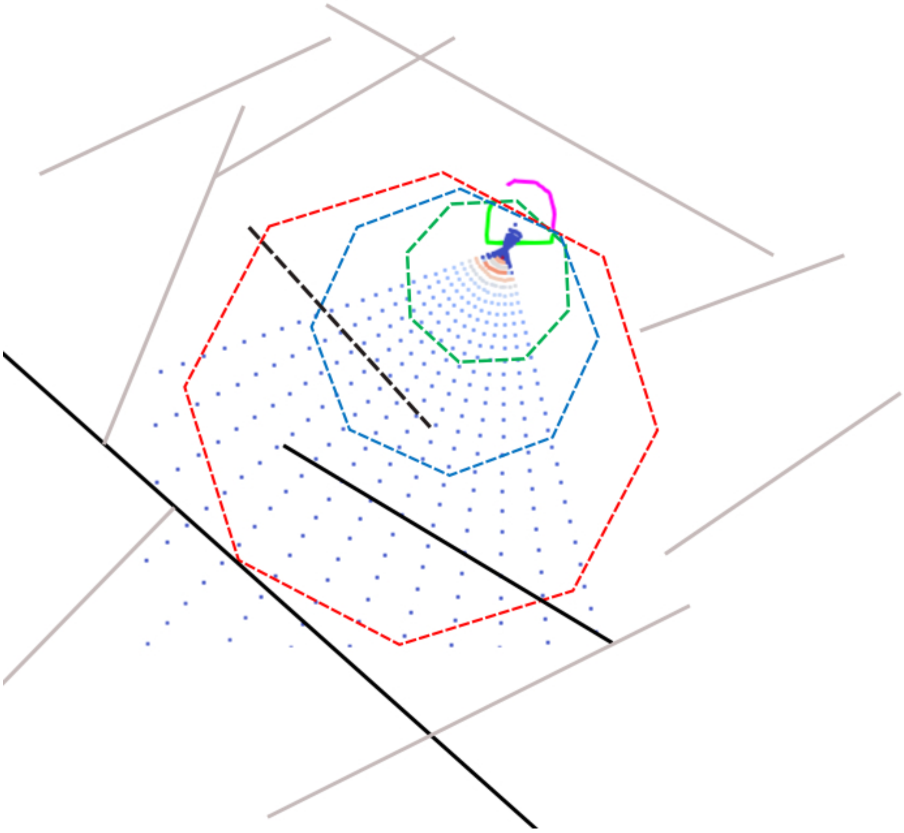


*Figure 4-2. Example illustrating the use of termination probability in the conditioning algorithm. The points show the possible centres, the octagons are three example possible fractures, and the grey and black lines are the intersects of pre-existing fractures with the current plane in which a conditioned fracture is to be grown.*

## 4.2 Algorithm demonstration Example

The algorithm is here demonstrated on a model with three tunnels and a single borehole with the same geometries as described in Section 3.2, see Figure 3-14. Target traces and borehole logs were generated using three G-DFN fracture sets with parameters presented in Table 4-1 and Table 4-2. Termination against other fractures was set to 100 %.

**Table 4-1. Fracture size distribution parameters and fracture intensity for 3 G-DFN sets.**

| Distribution | $r_0$ [m] | $k_r$ [-] | $r_{min}$ [m] | $r_{max}$ [m] | $P_{32}$ [m$^2$/m$^3$] |
|---|---|---|---|---|---|
| Power-Law | 2 | 2 | 2 | 100 | 0.065 |

**Table 4-2. Fracture orientation distribution parameters for 3 G-DFN sets.**

| Set | Distribution | Trend [deg] | Plunge [deg] | Concentration κ [-] |
|---|---|---|---|---|
| Set A | Fisher | 130 | 15 | 15 |
| Set B | Fisher | 60 | 15 | 15 |
| Set C | Fisher | 270 | 55 | 10 |

The first unconditioned realisation of the model results in total 225 traces larger than 1 m and 24 borehole intersections. Resulting traces were used as a target for the conditioning algorithm and a new conditioned realisation grown with the same set definition parameters. In an ideal situation the results from the target model and conditioned model should be identical. Comparison of the results from the conditioned model can be seen in Table 4-3 and Figure 4-3.

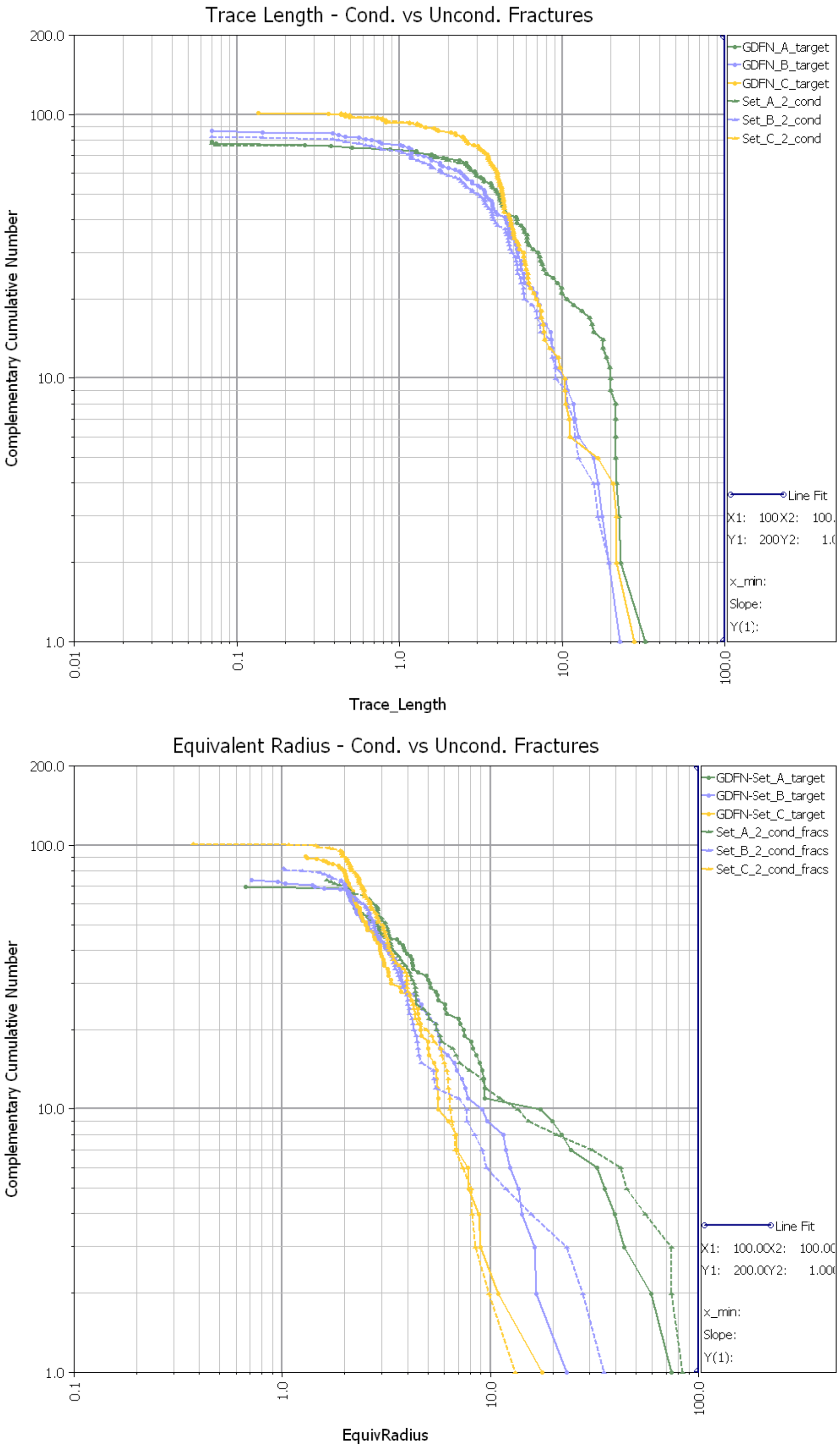


***Figure 4-3.*** *Comparison of tunnel trace length (upper) and size of tunnel intersecting fractures (lower) for a three set G-DFN: Sets A, B and then C are generated in sequence.*

It can be seen that consistent increases in slope occur in both trace and size with successively aged sets (A, B, C) for conditioned and unconditioned models. The conditioned model shows a good match on the number of traces and trace lengths both being within a few percent difference compared to target data. Comparing size of tunnel intersecting fractures, it can be observed that for Set A, which is generated first, the conditioned fractures tend to have slightly larger deviation than target fractures. Both other sets B and C show good match with the target fractures. Example unconditioned and conditioned G-DFN realisations are shown in Figure 4-4.

**Table 4-3. Comparison of conditioned and target models.**

| Trace/Logs | Count | Mean/std trace length [m] | Mean/std fracture radius [m] |
|---|---|---|---|
| Set A – target | 79 | 8.0/7.3 | 9.1/13.4 |
| Set A – conditioned | 78 | 8.0/7.3 | 9.6/17.2 |
| Set B – target | 87 | 5.0/4.5 | 4.9/4.2 |
| Set B – conditioned | 91 | 4.9/4.4 | 4.6/5.4 |
| Set C – target | 102 | 5.3/4.6 | 3.6/2.4 |
| Set C – conditioned | 107 | 5.4/4.6 | 3.6/2.0 |

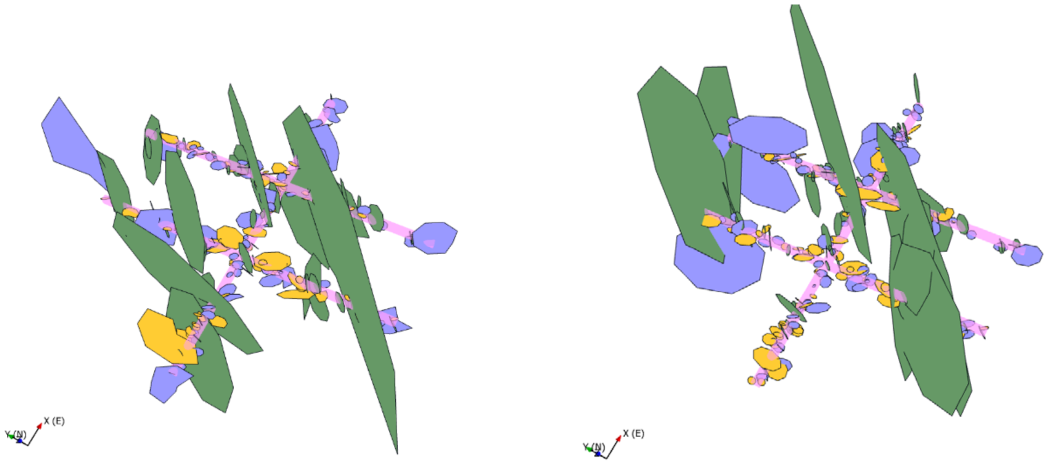


*Figure 4-4. A comparison of the unconditioned G-DFN (left) used to create the synthetic reality of traces and one realisation of the conditioned G-DFN (right).*

# 5 Conditioning of Semi-Deterministic Non-Planar Surfaces

During the ongoing project of development of a Geological Modelling Methodology (GMM, Hermanson and Petersson 2022), an additional requirement for conditioning the larger fractures observed in tunnels, such as critical structures rank 3 (CS3)(Critical structures 3, Munier and Mattila 2015), has arisen. During construction, new geological information will be acquired along the accesses, shafts and repository tunnels. The identified geological structures would be observed at a limited number of locations, which requires systematic rules for the interpretation. In GMM these rules specify for example how far from an observation point a surface representing a geological structure might extend or how to extend the surface above and below repository tunnels. An example illustrating such rules is shown Figure 5-1. In the following text these surfaces are referred to as “semi-deterministic” as part of the structure is defined deterministically following the GMM rules.

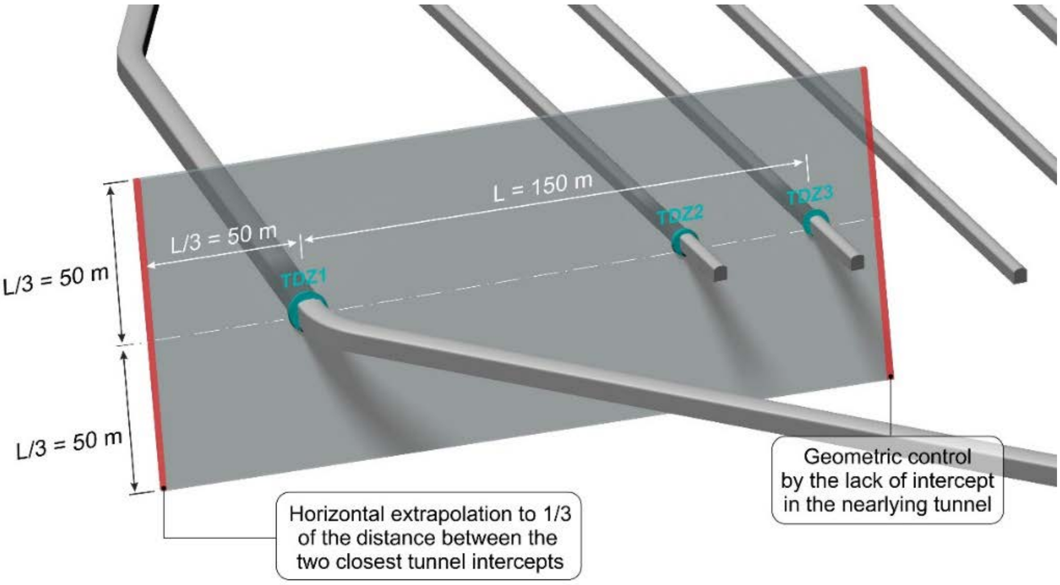

L = 150 m
L/3 = 50 m
TDZ1
TDZ2
TDZ3
L/3 = 50 m
L/3 = 50 m
Horizontal extrapolation to 1/3 of the distance between the two closest tunnel intercepts
Geometric control by the lack of intercept in the nearlying tunnel

***Figure 5-1.*** *Illustration of rules for interpreting a deformation zone observed at repository depth. From GMM 2020, Hermanson and Petersson 2022).*

From a DFN modelling perspective, there is a need to include these identified structures and extrapolate the input surfaces probabilistically outside the defined boundary in a manner that is consistent with the DFN recipe and at the same time respects all observations. Essentially this is a conditioning problem as solved in Chapters 2 and 4, but with extra constraints given by the edges of the geologist's interpreted structure.

The requirements for the conditioning are summarised below:

- the conditioned fracture must encapsulate the whole input surface,
- the generated fracture must follow the specified size distribution,
- the fracture should honour the specified terminations against deterministic objects,
- the fracture should not intersect other sampling objects where no observations were identified, and
- input surfaces can be both planar and non-planar.

This feature was designed to be generalised to work for non-planar surfaces. At this stage, the G-DFN algorithm is limited to planar fractures, and so these two capabilities cannot currently be combined.

## 5.1 Algorithm Workflow

The algorithm used to solve the above requirements utilises most of the modules developed for conditioning of statistical and G-DFN models. The algorithm workflow is as follows:

1) Extrapolate input surface.
2) Identify positive and negative constraints by calculating the intersections between the extrapolated surface and sampling objects (boreholes and tunnels).
3) Create a probability grid using a defined size distribution.
4) Sample a centre from the probability grid and generate conditioned fracture.
5) Identify potential terminations against other deterministic structures and if necessary, clip the resulting conditioned fracture.

These steps are detailed below.

### 5.1.1 Extrapolate input surface

One of the requirements for conditioning is that the input surfaces representing CS3 structures might not be planar, which introduces an additional complexity to the extrapolation process. For obtaining acceptable results from extrapolation, the input surface must meet certain requirements. The first requirement is that the surfaces must be clipped by a planar surface resulting in straight outline boundary. The second requirement is that for the algorithm to handle non-planar surfaces, the surface deviation from a plane should be relatively small for obtaining satisfactory results. This requirement is however reasonable for geological structures.

The algorithm first extracts the input surface outline and simplifies the polyline, so it contains only segments with different orientations. This step removes unnecessary vertices while still maintaining the polyline geometry. In next steps, the algorithm loops through all outline vertices and computes extrapolation vectors which are pointing outside the surface as shown on Figure 5-2. The extrapolation vector is computed as a half angle, $\beta$, between vectors $v_1$ and $v_2$ projected into a plane with a normal defined as the cross-product of $v_1$ and $v_2$. In the next step, a new outline is formed by offsetting all input outline vertices in the direction of extrapolation vectors. The new surface is that formed by merging the original surface together with triangulation between the two outlines.

For surfaces with high undulation or not well-defined edges the shape of the extrapolated surface might differ from the input. For reducing this boundary disturbance effect, it is better to use larger extrapolation distance. In the current implementation, the extrapolation distance is equal to $2 \times R_{max}$, where $R_{max}$ represents the maximal fracture radius of the input surface.

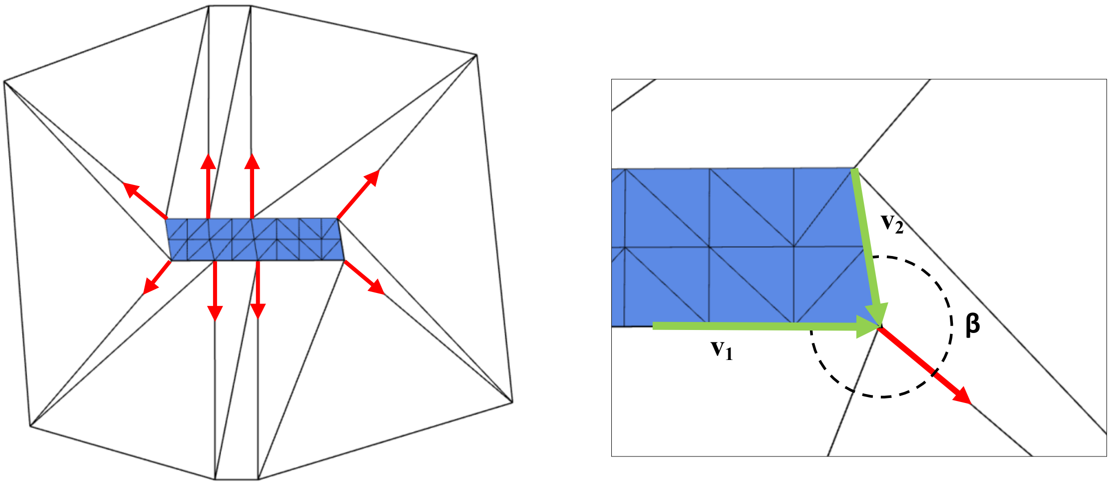


***Figure 5-2.*** *Showing extrapolation vectors (red arrows) for the input surface (left) and detail with definition of an extrapolation vector (right).*

### 5.1.2 Identify positive and negative constraints and create a probability density grid

Conditioning of semi-deterministic CS3 surfaces is made using similar algorithms as presented in Section 2.1. For construction of the probability grid, the algorithm first needs to define positive and negative constraints. The outline of the input surface is considered as the only positive constraint (note that this is always made up of FPI constraints). Negative constraints are defined by intersections between the extrapolated surface and all sampling objects which are not intersected by the input surface itself.

Both positive and negative constraints are projected into a 2D plane, and the probability density grid is created. To be able to calculate weights of the probability centre points, it is required that a user defines a size distribution for the conditioned CS3 objects. Stochastic DFN modelling of deformation zone midplanes for a site would deliver such a model, based on deterministic modelling or interpretations of surface lineaments.

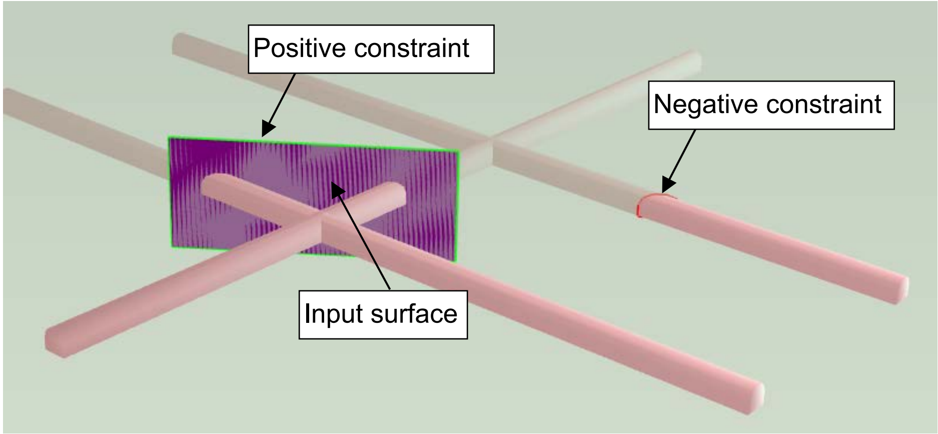


***Figure 5-3.*** *Showing semi-deterministic input surface and positive (green) and negative (red) constraints.*

### 5.1.3 Fracture generation

A fracture centre and fracture size are sampled from the probability density grid and the algorithm checks if all positive and negative constraints are fulfilled. In the next step, a cylindrical surface with a size of the sampled fracture radius is placed at the sampled fracture centre. The extrapolated surface is then clipped by this cylindrical surface and converted to a tessellated fracture, as shown in Figure 5-4.

### 5.1.4 Terminations

The semi-deterministic conditioning algorithm also allows to simulate terminations defined in the input surfaces. To define a termination for a specific input surface, it is necessary to clip the input surface against other surfaces, which in practice would be formed by a deterministic geological object, such as a large deformation zone as illustrated in Figure 5-5. For each input surface, the algorithm looks for all touching deterministic surfaces which are then used to clip the resulting conditioned fracture.

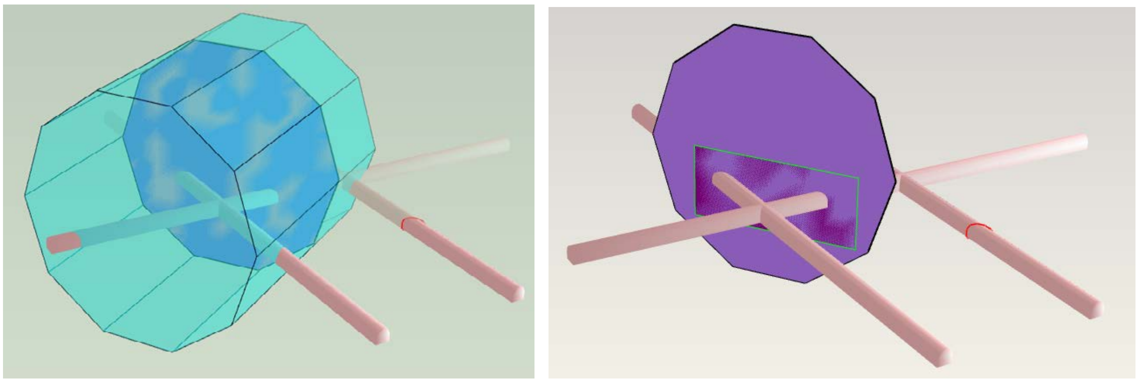


***Figure 5-4.*** *Cylindrical surface used for clipping the extrapolated surface (left) and resulting conditioned fracture (right).*

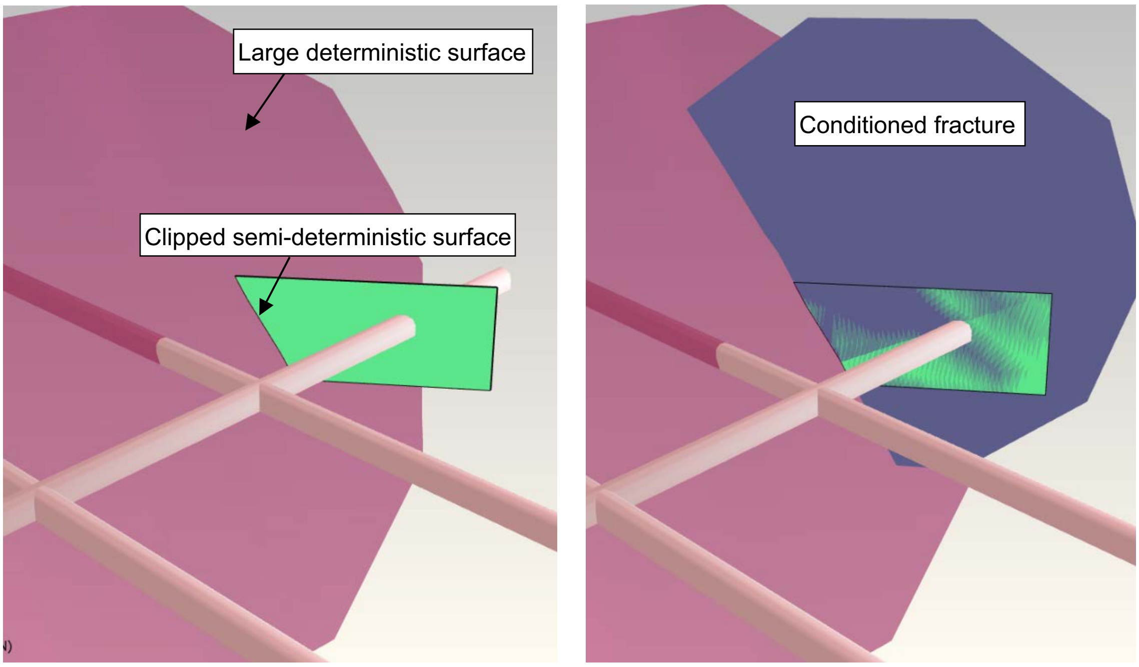

Large deterministic surface
Clipped semi-deterministic surface
Conditioned fracture

***Figure 5-5.*** *Semi-deterministic input surface clipped against a deterministic surface (left) and the resulting conditioned fracture (right).*

## 5.2 Example

Semi-deterministic conditioning is demonstrated on a model with five deposition tunnels and two boreholes. The model contains 7 CS3 surfaces both planar and non-planar. Some of the semi-deterministic surfaces are terminated against the large deterministic structures, see Figure 5-6. A power-law size distribution ($r_0$ = 50 m, $k_r$ = 2.6) for the stochastic DZ model was used in the conditioning.

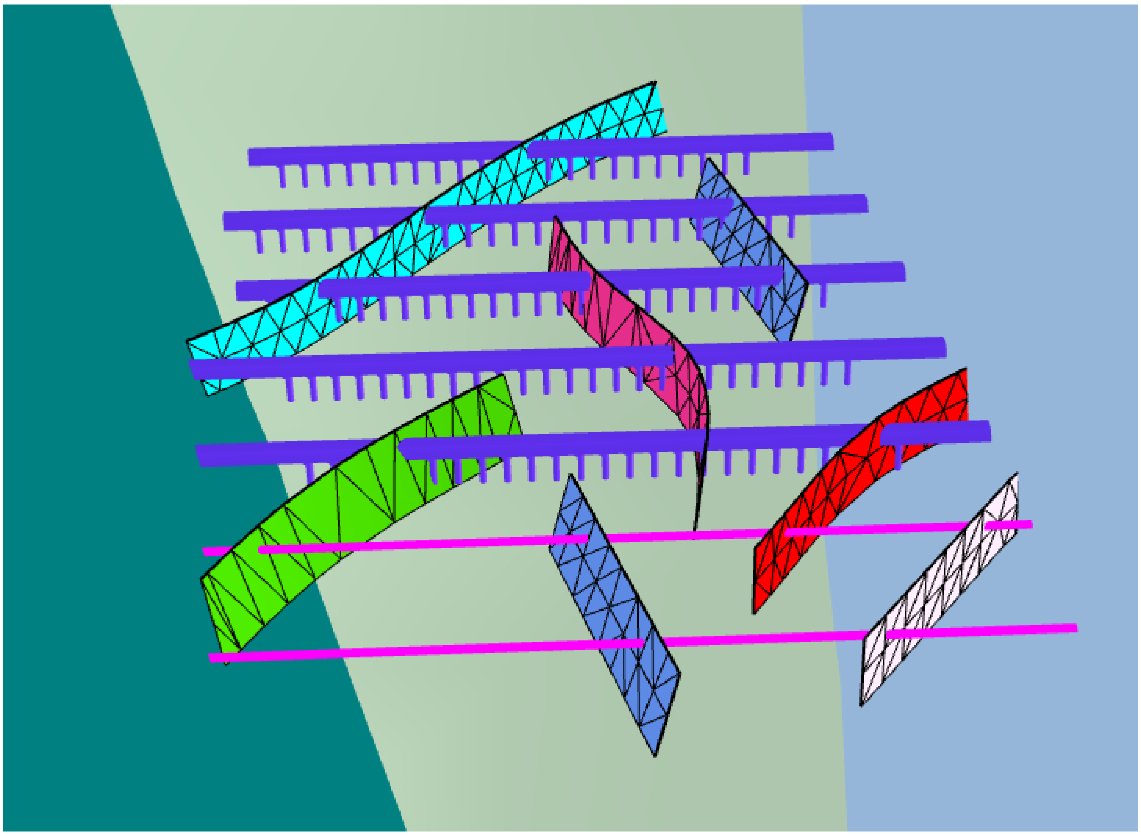


***Figure 5-6.*** *Example model with 5 deposition tunnels and two boreholes enclosed by three large deterministic faults. The model contains 7 semi-deterministic planar and non-planar surfaces.*

The results of three example conditioned realisations are presented in Figure 5-7. Each conditioned realisation meet al. requirements specified in Chapter 5. The algorithm is able to produce both planar and non-planar conditioned fractures which intersect the sampling objects only at observed locations (positive constraints), follow the specified size distribution as well as respecting specified terminations against deterministic surfaces.

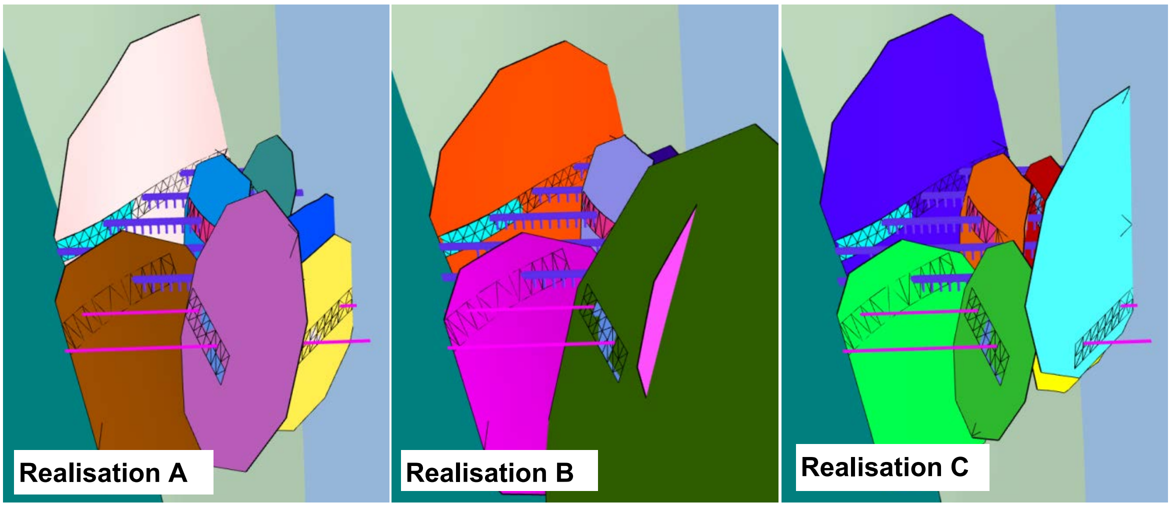


***Figure 5-7.*** *Three realisations of the semi-deterministic conditioning.*

# 6 Conclusions

A new method to condition realisations of a fracture network to honour observed traces on surfaces (tunnels and outcrops) and borehole intersects has been implemented in FracMan. The new method has several advantages over conditioning algorithms implemented in previous software versions. Firstly, it supports spatial variations in fracture intensity across the model, e.g., where intensity is a function of depth, lithology or proximity to deformation zones. These spatial variations would normally be defined through use of a geocellular model. The conditioned realisations will attempt to preserve the spatial model defined in the stochastic fracture recipe away from the observation. It also ensures that the full range of possible fractures fitting each trace can be realised over an ensemble of realisations. It achieves this by constructing the specific probability density function of all possible fractures that conform with each individual mapped trace and then samples one of them. The method is fast and capable of conditioning many realisations of large DFN models to many thousands of observed fracture traces on regular modelling laptop. It should be noted that for G-DFN models the conditioning algorithm doubles the modelling time due to the need to obtain size distribution of intersected fractures.

The conditioning workflow is performed in two steps. The first part of the algorithm constructs a grid of points corresponding to all centres of possible fractures that can meet the constraints implied by a particular trace, or set of traces interpreted to belong to the same structure. The second part of the algorithm calculates the probability that the conditioned fracture has a centre at each point in the grid, noting that there can be multiple fractures with different sizes and rotations centred on that point and complying with the observed traces. The probabilities are calculated from the specified fracture size distribution for the fracture set being conditioned, and its shape, meaning that the size distributions of the fracture sets being conditioned are key inputs to the conditioning process. Hence, an essential precursor is to interpret a reliable intensity-size scaling model for fractures of the scales that are being conditioned to.

Conditioning is only done on fracture geometries. However, the detection of flow associated to a particular trace can be set as an extra constraint whereby it is required that the fracture conditioned on that trace is connected to the background network and chosen boundaries of the model. This at least ensures that the conditioned fracture has the potential to flow, connectivity being pre-requisite for flow.

When using real data, the importance of pre-processing mapping/logging data to provide a description of the traces that reflects a geological realistic interpretation of the traces cannot be over-emphasised. This means linking traces across mapping surfaces and objects where they are believed to be expressions of the same structure. Some tools have been included to aid this process, but more tools are probably needed to provide a complete workflow for picking traces and making the necessary associations between them. The method has been tested on real data from the ONKALO (Hartley et al. 2021).

This basic algorithm for creating conditioned DFNs for statistical fracture recipes has been extended to its application on grown DFNs whereby fractures are grown onto the observed traces and beyond them in the case of FPI fractures. In this case, the probability density function for centres and sizes of conditioned fractures depends not only the specified fracture recipe, but also the background network that exists at the time each trace is conditioned with a grown fracture. Although this is a complex situation, the results appear acceptable.

Further, the same algorithm has also been adapted to the challenge of producing conditioned realisations of fractures/deformation zones around planar and non-planar geologically interpreted semi-deterministic structures interpreted consistent with an underlying statistical model. This allows the probabilistic assessment of the effects of CS3 structures, for example.

# References

SKB's (Svensk Kärnbränslehantering AB) publications can be found at www.skb.com/publications.

**Bym T, Hermanson J, 2018.** Methods and workflow for geometric and hydraulic conditioning. DFN-R – status report 2013–2015. SKB R-17-12, Svensk Kärnbränslehantering AB.

**Candela T, Renard F, Bouchon M, Brouste A, Marsan D, Schmittbuhl J, Voisin C, 2009.** Characterization of Fault Roughness at Various Scales: Implications of Three-Dimensional High Resolution Topography Measurements. Pure and Applied Geophysics 166, 1817–1851. https://doi.org/10.1007/s00024-009-0521-2

**Davy P, Le Goc R, Darcel C, 2013.** A model of fracture nucleation, growth and arrest, and consequences for fracture density and scaling. Journal of Geophysical Research: Solid Earth 118, 1393–1407. https://doi.org/10.1002/jgrb.50120

**Golder, 2022.** FracMan v 8.0 manual. Bourne End: Golder Associates Ltd.

**Hartley L, Baxter S, Bym T, Cottrell M, Libby S, 2021.** Exploring the Effects of Tunnel Construction on the Hydraulic Acceptance of Deposition Holes. Working Report 2021-17, Posiva Oy, Finland.

**Hermanson J, Petersson J, 2022.** Methodology for deterministic geological modelling. Designed for the development of the Forsmark deep repository for spent nuclear fuel. R-20-10, Svensk Kärnbränslehantering AB.

**Libby S, Hartley L, Turnbull R, Cottrell M, 2019.** Grown Discrete Fracture Networks: a new method for generating fractures according to their deformation history. ARMA 19–1559 American Rock Mechanics Association.

**Munier R, 2010.** Full perimeter intersection criteria. Definitions and implementations in SR-Site. SKB TR-10-21. Svensk Kärnbränslehantering AB.

**Munier R, Mattila J, 2015.** Memo on Terminology. POS-021685, Posiva Oy, Finland.

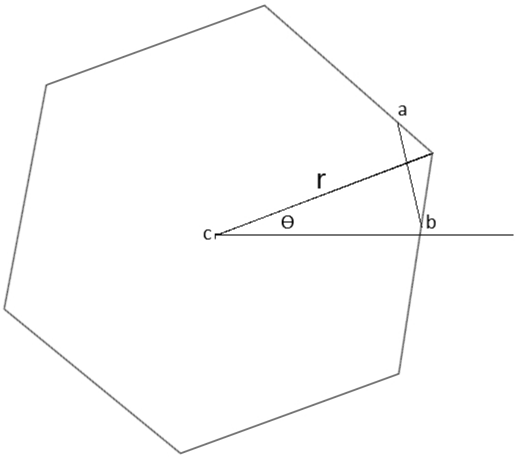
**Selroos J-O, Ivars D M, Munier R, Hartley L, Libby S, Davy P, Darcel C, Trinchero P, 2022.** Methodology for discrete fracture network modelling of the Forsmark site. Part 1: Concepts, Data and Interpretation Methods. SKB R-20-11, Svensk Kärnbränslehantering AB.

**Stigsson M, 2016.** Orientation Uncertainty of Structures Measured in Cored Boreholes: Methodology and Case Study of Swedish Crystalline Rock. Rock Mechanics and Rock Engineering 49, 4273–4284. https://doi.org/10.1007/s00603-016-1038-5

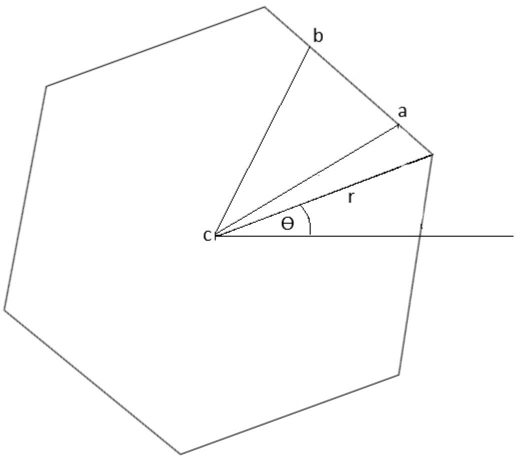
## Polygon Fitting Calculation

We are given two points representing the end points of a trace and a point representing the centre of the fracture, and wish to calculate possible values for the radius and rotation of a regular polygon with a given number of sides and the given centre such that the two trace end points lie on the perimeter of the polygon.

Let the two trace end points be **a** and **b**, and the desired centre for the polygon **c**. We will assume that **a**, **b** and **c** are non-colinear so that they define a unique plane containing the polygon, and that we have already chosen a coordinate system so that the polygon is in the xy plane. Let *n* be the (known) number of sides that defines the polygon. Let us call the (unknown) 'radius' of the polygon *r*, by which we mean the distance from the centre **c** to one of the vertices. And similarly, the unknown 'rotation' of the polygon *Θ*, by which we mean the angle between the positive x axis and a line joining **c** to one of the polygon's vertices chosen so that *Θ* is in the range [0, 2 π/*n*).

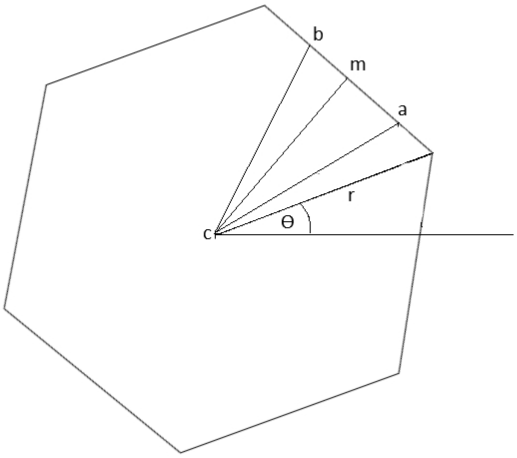


If we think of the two end points as being constraints on the possible values for the radius, it is clear that we can rotate those points about **c** through any multiple of 2 π/*n* without changing those constraints. We can thus always reduce it to a case where **a** and **b** both lie on the same side of the polygon.



Bearing in mind that to have **a** and **b** both on the same polygon edge the angle acb can be at most $2\,\pi/n$, we get two possibilities to try. A convenient way to find those is to rotate both **a** and **b** so that they are in the range $[0, 2\,\pi/n)$, which gives one of the possibilities; and then rotate the one with the smaller angle by $2\,\pi/n$ to give the second possibility. These two possibilities can also be thought of as having either an odd or an even number of polygon vertices in between the original points **a** and **b**.

As the points **a** and **b** are now on the same edge of the polygon we can easily find the midpoint **m** of that edge (if **u** is a unit vector in the direction $(\mathbf{b}-\mathbf{a})$ then $\mathbf{m} = \mathbf{a} - ((\mathbf{a}-\mathbf{c})\mathbf{u})\mathbf{u}$). And thus the rotation $\Theta$ by adding $\pi/n$ to the angle that $(\mathbf{m}-\mathbf{c})$ makes with the x axis and then reducing to a value in the range $[0, 2\,\pi/n)$. Furthermore $r = \frac{|m-c|}{cos(\pi/n)}$.



Although this gives us values for $r$ and $\Theta$ we must check that the points **a** and **b** do indeed lie on the perimeter of the polygon and not on the extension of a side. This can be achieved by checking that the distances from **c** to **a** and **b** are within the permitted range [|**m** – **c**|, $r$] or that the distance from **m** to **a** and **b** are at most half the length of the polygon's side.

Note that there is also a special case where **c** is on the perpendicular bisector of **ab** and the angle acb is an integer multiple of $2\,\pi/n$. In this case there are infinitely many possible solutions for $r$ and $\Theta$. As with the case where **a**, **b**, and **c** are colinear we ignore this possibility. It manifests itself as the case where the rotated points **a** and **b** coincide, which needs to be eliminated to avoid division by zero in the calculation. In practice this is unlikely to occur with randomly chosen points **c**. Nevertheless, there will be points arbitrarily close to such points **c** where the calculation above yields possible polygons.

Arkitektkopia AB, Bromma, 2024

SKB is responsible for managing spent nuclear fuel and radioactive waste produced by the Swedish nuclear power plants such that man and the environment are protected in the near and distant future.

**skb.se**

SVENSK KÄRNBRÄNSLEHANTERING