R-09-48

Statistics of modelled conductive fractures based on Laxemar and Forsmark Site descriptive model data

Martin Stigsson, Svensk Kärnbränslehantering AB

December 2009

Svensk Kärnbränslehantering AB
Swedish Nuclear Fuel
and Waste Management Co

Box 250, SE-101 24 Stockholm
Phone +46 8 459 84 00

ISSN 1402-3091
SKB Rapport R-09-48

Statistics of modelled conductive fractures based on Laxemar and Forsmark Site descriptive model data

Martin Stigsson, Svensk Kärnbränslehantering AB

December 2009

Keywords: DFN, SER, Transmissivity, Statistical distribution, Scanline, Sampling.

A pdf version of this document can be downloaded from www.skb.se.

Abstract

The objectives of this report is to investigate the frequency of fractures assumed to be water conductive, i.e. open or partly open and directly or indirectly connected to a source. Also the distribution of total transmissivity in 100 m and 20 m horizontal sections and 8 m vertical sections is calculated.

The report is only intended to serve as input to the SER, Site Engineering Report /SKB 2008a, b/, at Laxemar and Forsmark. The input data for the analyses is taken, as is, from the Discrete Fracture Network sections in /Rhén et al. 2008/ and /Follin et al. 2007a, b/. No evaluation that the model parameters are appropriate for the task or sensitivity analysis is performed.

The tunnels and deposition holes are modelled as scanlines which is a very coarse approximation, but it may give some rough estimation of the frequency of the water bearing features, especially for the larger ones, and the total transmissivity in a section.

Sammanfattning

Syftet med denna rapport är att undersöka frekvensen av sprickor som antas vara vattenförande, dvs helt eller delvis öppna och direkt eller indirekt i kontakt med någon källa. Även fördelningen av total transmissivitet i 100 m och 20 m horisontella samt 8 m vertikala sektioner beräknas.

Rapporten är endast avsedd att användas som indata till SER, Site Engineering Report /SKB 2008a, b/, i Laxemar and Forsmark. Indata för analysen är hämtad rakt av från diskreta spricknätverksavsnitten i /Rhén et al. 2008/ och /Follin et al. 2007a, b/. Ingen utvärdering har gjorts att parametrarna är ändamålsenliga för uppgiften inte heller någon sensitivitetsanalys har utförts.

Tunnlarna och deponeringshålen är modellerade som linjer vilket är en mycket förenklad approximation, men det bör kunna ge en grov uppskattning av frekvensen av vattenförande sprickor, speciellt för de stora, och den totala transmissiviteten i en sektion.

Contents

1 Introduction

The current work was initiated to give a coarse picture of the need for grouting deposition tunnels and holes for a KBS-3 repository at Laxemar and Forsmark.

1.1 Objectives

The objectives of this report is to investigate the frequency of fractures assumed to be water conductive, i.e. open or partly open and directly or indirectly connected to a source, together with the distribution of total transmissivity in sections of different lengths. The transmissivity distribution of fractures intersecting a tunnel or a deposition hole modelled as a scan line is calculated using numerical methods.

1.2 Scope

The results are only intended to serve as input to the SER, Site Engineering Reports /SKB 2008a, b/, at Laxemar and Forsmark. The input data for the analyses is taken, as is, from the Discrete Fracture Network sections in /Rhén et al. 2008/ and /Follin et al. 2007a, b/. No evaluation that the models parameters are appropriate for the task or sensitivity analysis is performed.

Modelling a tunnel or deposition hole as a scanline is a very coarse model, especially since the DFN, Discrete Fracture Network, model characterizes fractures with radii from 0.28 m. But it may give some rough estimation of the frequency of the water bearing fractures that one can expect, especially for the larger fractures with radii larger than some meters.

The nomenclature of rock volumes assumed having different parameters of hydraulic active fractures is different between Laxemar and Forsmark. In Laxemar the volumes are called Hydraulic Rock Domains, HRD, whilst they coincide with the Fracture Domains in Forsmark and hence are called Fracure Domains Forsmark, FFM. In Laxemar three HRDs are modelled with parameters corresponding to the depth 400 to 650 m below sea level and in Forsmark two FFMs are modelled with parameters corresponding to the depth beneath 400 m below sea level.

2 Model set up

2.1 Model steps

The study consists of five steps,

1. generate fractures,
2. delete isolated fractures,
3. sample the remaining fracture network,
4. calculate raw statistics and
5. evaluate and visualize data.

Generation of fractures together with sorting of connected and none-connected fractures is performed using DarcyTools version 3.2 /Svensson 2008/. The sampling and the calculation of raw statistics are performed by two different console applications, written in Fortran95 (Intel Fortran 11) and Visual Basic (Visual studio .Net 2003) respectively, developed for this study only. A copy of the source codes to the applications are shown in Appendix A1 and A2. The evaluation and visualization is done in Excel 2007.

2.2 Rock volumes modelled

The report is limited to reflect the conditions at depth interval 400 to 650 m below sea level for the three HRDs, Hydraulic Rock Domains, named HRD_C, HRD_W and HRD_EW007 in Laxemar, see Figure 2-1, and beneath 400 m below sea level for the two fracture domains FFM01 and FFM06 in Forsmark, see Figure 2-2. In Forsmark the fracture domain FFM06 has limited amount of data and is given the same properties as the FFM01. Consequently there is no distinction between FFM06 and FFM01 in this report and hence both are included in the name FFM01 further on.

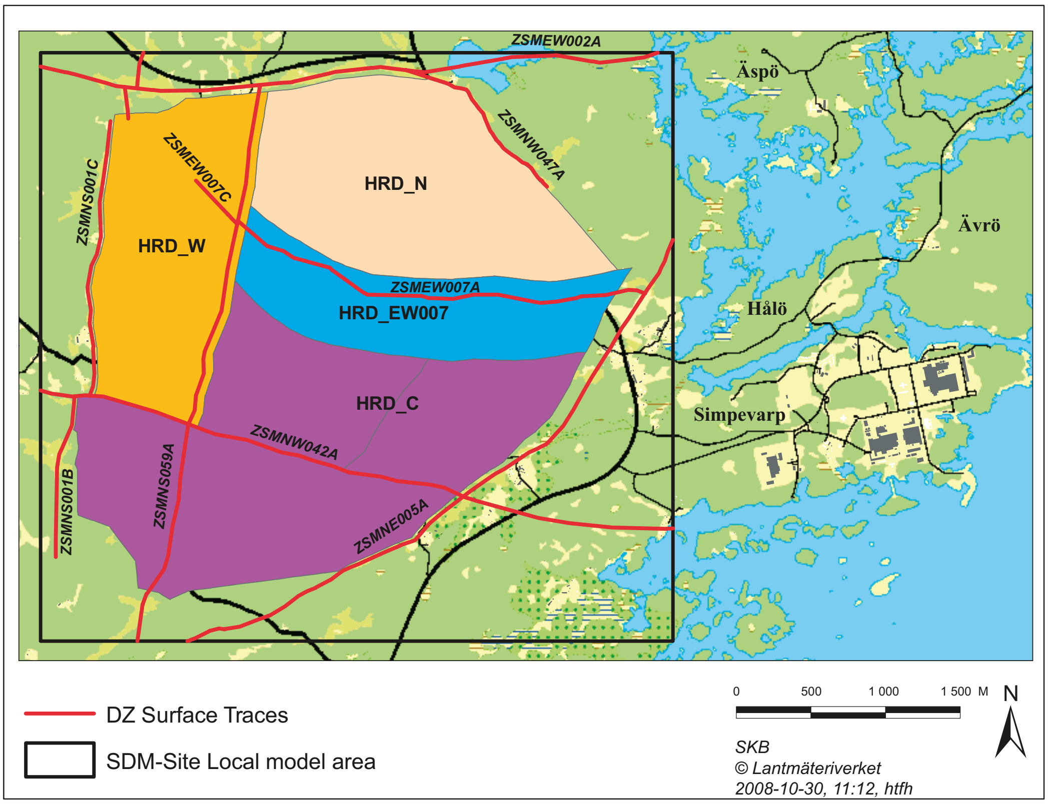

Figure 2-1. *Surface footprint of the Hydraulic Rock Domains, HRD, in Laxemar.*

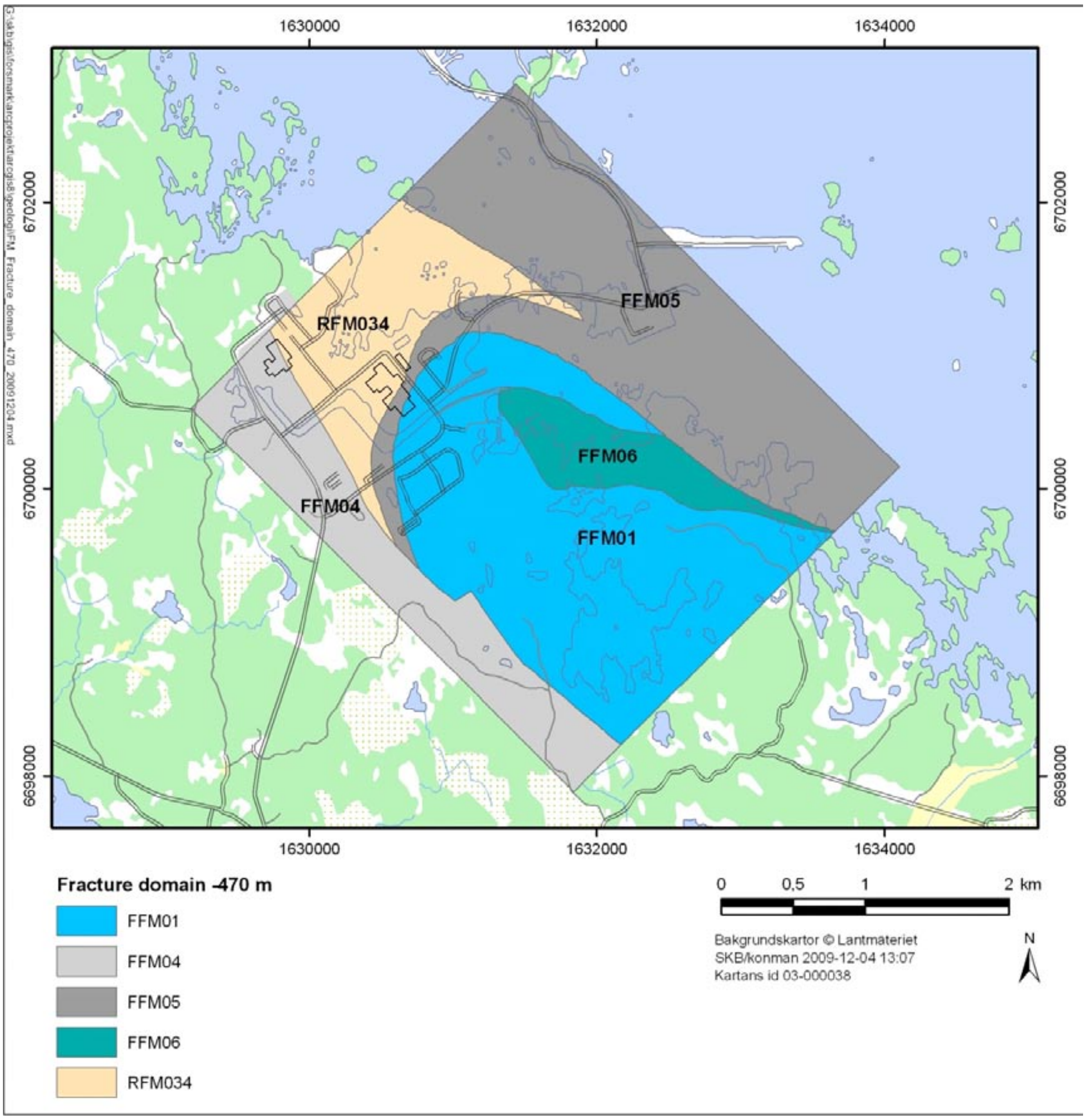

Figure 2-2. *Footprint at 470 m below surface of the fracture domains, FFMs, in Forsmark.*

The orientation of the deposition tunnel are WNW/ESE in Laxemar, more specifically 130° CW/50° CCW rotation from north to reflect the assumed orientation of a repository layout at Laxemar /SKB 2008b, Hansson et al. 2009b/. The deposition tunnels in Forsmark is also supposed be oriented WNW/ESE (140° ±15°) /SKB 2008a, Hansson et al. 2009a/, hence the orientation of tunnels is assumed 140° CW/40° CCW.

2.3 Fracture model

The fractures are modelled using DarcyTools 3.2 /Svensson 2008/ which implies that the fractures are assumed to be square with a thickness. To make a good approximation of discrete features without the risk of getting into numerical problems all fractures, regardless the size, are modelled with 1 mm thickness.

2.3.1 Laxemar

The fracture models for potential flowing features, i.e. open and partly open, are taken from Tables 10-7, 10-12 and 10-15 in /Rhén et al. 2008/, and the extracted data used in this study are repeated below as Table 2-1 to Table 2-3.

Table 2-1. Fracture parameters for HRD_C at depth 400 to 650 m below sea level.

Set	Trend (°)	Plunge (°)	Fisher conc (–)	k_r (–)	r_0 (m)	$P_{32(0.038–564)}$ (m^2/m^3)
ENE	155.1	3.4	9.6	2.80	0.038	0.38
WNW	204.1	1.6	12.0	2.50	0.038	0.74
N-S	270.2	8.4	7.8	2.90	0.038	0.47
SubH	46.3	84.7	12.0	2.90	0.038	0.58

Table 2-2. Fracture parameters for HRD_W at depth 400 to 650 m below sea level.

Set	Trend (°)	Plunge (°)	Fisher conc (–)	k_r (–)	r_0 (m)	$P_{32(0.038–564)}$ (m^2/m^3)
ENE	340.3	1.2	15.0	2.80	0.038	0.17
WNW	208.9	2.2	10.9	2.55	0.038	0.33
N-S	272.8	12.0	11.5	2.55	0.038	0.30
SubH	277.1	84.3	11.1	2.65	0.038	0.38

Table 2-3. Fracture parameters for HRD_EW007 at depth 400 to 650 m below sea level.

Set	Trend (°)	Plunge (°)	Fisher conc (–)	k_r (–)	r_0 (m)	$P_{32(0.038–564)}$ (m^2/m^3)
ENE	162.8	1.4	10.7	2.95	0.038	0.69
WNW	25.3	0.2	16.4	2.50	0.038	1.43
N-S	88.9	3.9	8.8	2.95	0.038	0.64
SubH	138.7	81.3	9.7	2.95	0.038	0.92

According to /Rhén et al. 2008/ the smallest and largest fracture radius generated when calibrating the transmissivity model, see further in section 2.4, is 0.282 m and 564 m respectively. A visualisation of the modelled complementary cumulative fracture intensities in this range for the three hydraulic rock domains at the studied depth are shown in Figure 2-3.

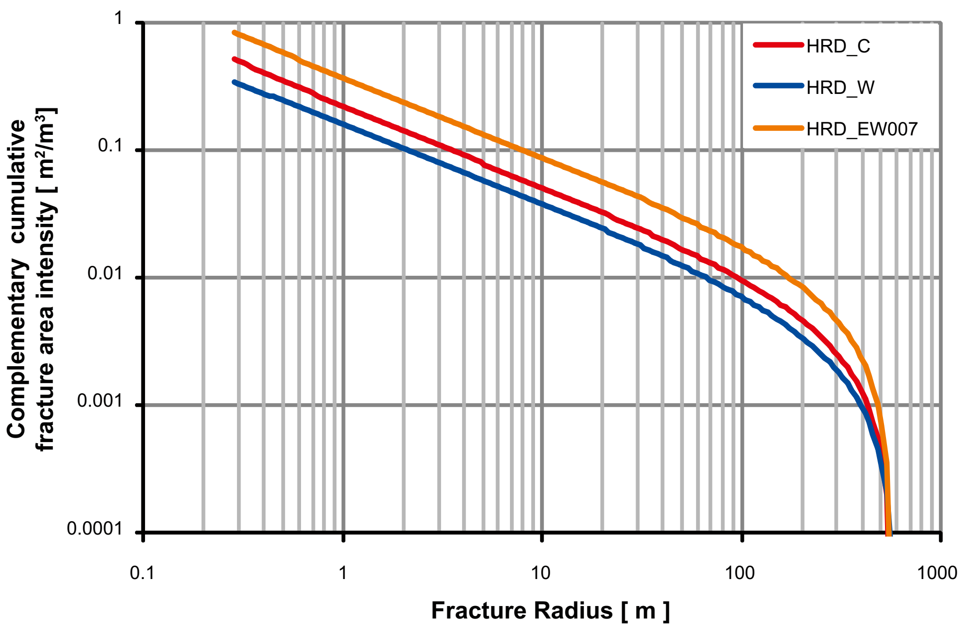

Figure 2-3. *The Modelled complementary cumulative fracture area intensity for the three modelled HRDs, i.e. the total modelled fracture area per volume of rock for fractures larger than a specific radius.*

2.3.2 Forsmark

The fracture models for potential flowing features, i.e. open and partly open, are taken from Table 3-7 in /Follin et al. 2007a/. The extracted data used in this study are repeated below as Table 2-4 and illustrated in Figure 2-4.

Table 2-4. Fracture parameters for FFM01 at depth beneath 400 m below sea level.

Set	Trend (°)	Plunge (°)	Fisher conc (–)	k_r (–)	r_0 (m)	$P_{32(0.038–564)}$ (m^2/m^3)
NS	87	2	21.7	2.50	0.038	0.094
NE	135	3	21.5	2.70	0.038	0.163
NW	41	2	23.9	3.10	0.038	0.098
EW	190	1	30.6	3.10	0.038	0.039
HZ	343	80	8.2	2.38	0.038	0.141

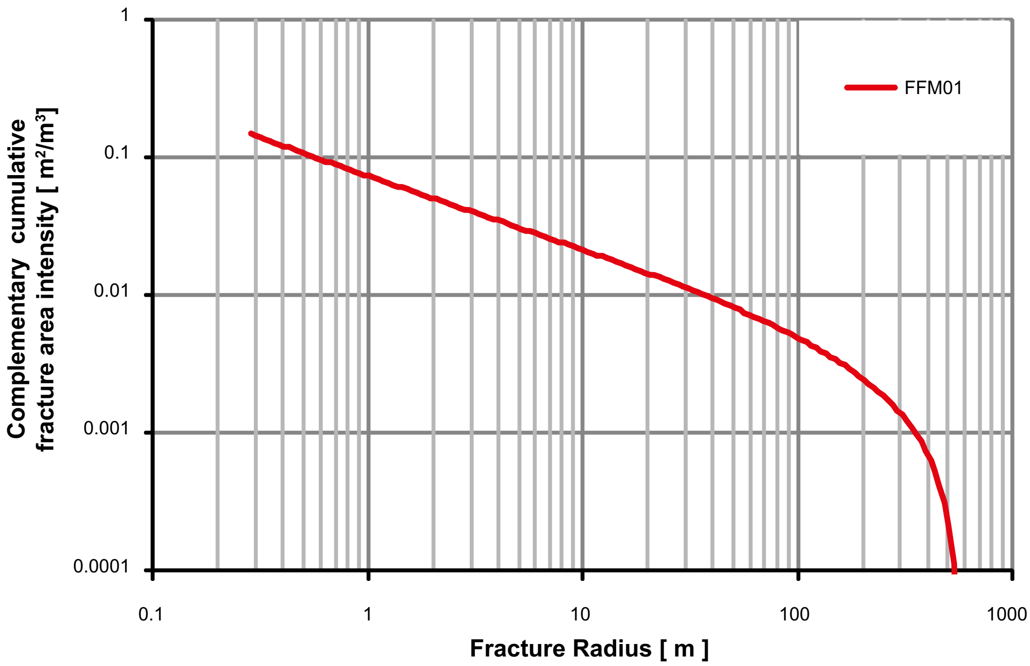

Figure 2-4. *The Modelled complementary cumulative fracture area intensity for FFM01, i.e. the total modelled fracture area per volume of rock for fractures larger than a specific radius.*

2.3.3 Transformation to DarcyTools parameters

The input parameters to DarcyTools are somewhat different to those stated in Table 2-1 to Table 2-4 and hence need to be calculated according to Formula 2-1 to Formula 2-6.

$$l_0 = r_0 \cdot \sqrt{\pi}$$ Formula 2-1

$$I = P_{32(r_0-\infty)} \cdot l_0^{(k_r-2)} \cdot (k_r - 2) / l_{ref}^{k_r}$$ Formula 2-2

$$D = -k_r$$ Formula 2-3

$$\lambda_1 = \cos(90 - tr) \cdot \cos(pl) \cdot \kappa$$ Formula 2-4

$$\lambda_2 = \sin(90 - tr) \cdot \cos(pl) \cdot \kappa$$ Formula 2-5

$$\lambda_3 = -\sin(pl) \cdot \kappa$$ Formula 2-6

Where

l_0	=	side of the smallest fracture modelled represented as a quadrate [m]
r_0	=	radius of the smallest fracture modelled represented as a disc [m]
I	=	intensity of fractures per volume of a reference size [m^{-3}]
$P_{32(a-b)}$	=	fracture area per volume for fractures in the size range a to b [m^2/m^3]
l_{ref}	=	reference length for the intensity [m]
k_r	=	shape parameter if applied on the denominator [–]
D	=	shape parameter if applied on the numerator [–]
λ_{1-3}	=	set mean pole vectors including the Fisher concentration [–]
tr	=	trend [°]
pl	=	plunge [°]
κ	=	Fisher concentration [–]

The relationship between the fracture intensities $P_{32(r0-564)}$, as stated in Table 2-1 to Table 2-4, and $P_{32(0.282-\infty)}$ as needed to calculate the intensity for use in DarcyTools is

$$P_{32(r_a-\infty)} = P_{32(r_b-r_c)} \cdot \frac{r_a^{(2-k_r)}}{r_b^{(2-k_r)} - r_c^{(2-k_r)}}$$ Formula 2-7

2.4 Transmissivity model

In /Rhén et al. 2008/ and /Follin et al. 2007b/ there are provided three different models for assigning transmissivity to a fracture, the correlated, the semi correlated and the uncorrelated model. The uncorrelated model decouples transmissivity from size, whilst the correlated is the other extreme where a specific radius always gives the same transmissivity. The semi correlated model has one part where transmissivity grow with size, but also a random part that can change the transmissivity within a given span. Only the correlated, described by Formula 2-8, and the semi correlated, described by Formula 2-9, models are regarded in this study.

$$T = a \cdot r^b$$ Formula 2-8

$$\log_{10} T = \log_{10}(a \cdot r^b) + \sigma \cdot N(0,1)$$ Formula 2-9

Where

T	=	transmissivity [m^2/s]
a	=	location parameter [m^2/s]
b	=	shape parameter [–]
r	=	radius of fracture [m]
σ	=	standard deviation [–]
$N(0,1)$	=	normal distribution with average 0 and standard deviation 1 [–]

2.4.1 Laxemar

Table 2-5 to Table 2-7 show parameters for the relationship between fracture size and transmissivity for the three studied HRDs at the depth 400 to 650 m below sea level. The tables are extractions of Tables 10-7, 10-12 and 10-15 in /Rhén et al. 2008/. Visualisations of the relationships for each fracture set in each HRD are shown in Figure 2-5 to Figure 2-16.

Table 2-5. Parameters for the relationship between transmissivity and radius in HRD_C at depth 400 to 650 m below sea level. Excerption from Table 10-7 in /Rhén et al. 2008/.

Set	Correlated		Semi correlated		
	a	b	a	b	σ
ENE	$3 \cdot 10^{-8}$	0.7	$5 \cdot 10^{-7}$	0.5	0.5
WNW	$3 \cdot 10^{-9}$	0.9	$2 \cdot 10^{-8}$	0.6	0.4
N-S	$1 \cdot 10^{-8}$	0.5	$1 \cdot 10^{-8}$	0.4	0.4
SubH	$1.5 \cdot 10^{-7}$	0.9	$3 \cdot 10^{-7}$	0.6	0.6

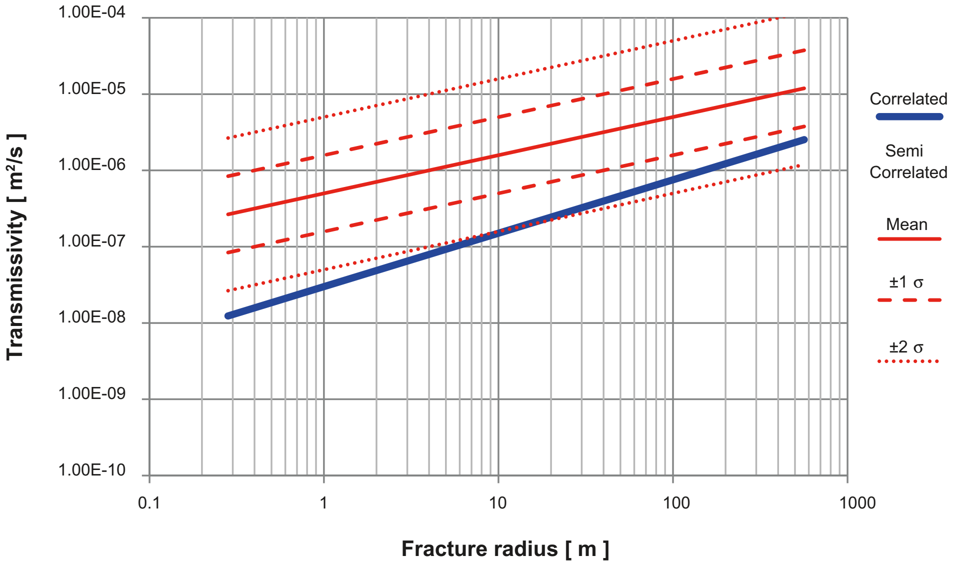

Figure 2-5. Relationship between radius and transmissivity for the ENE striking fracture set in HRD_C at depth 400 to 650 m below sea level.

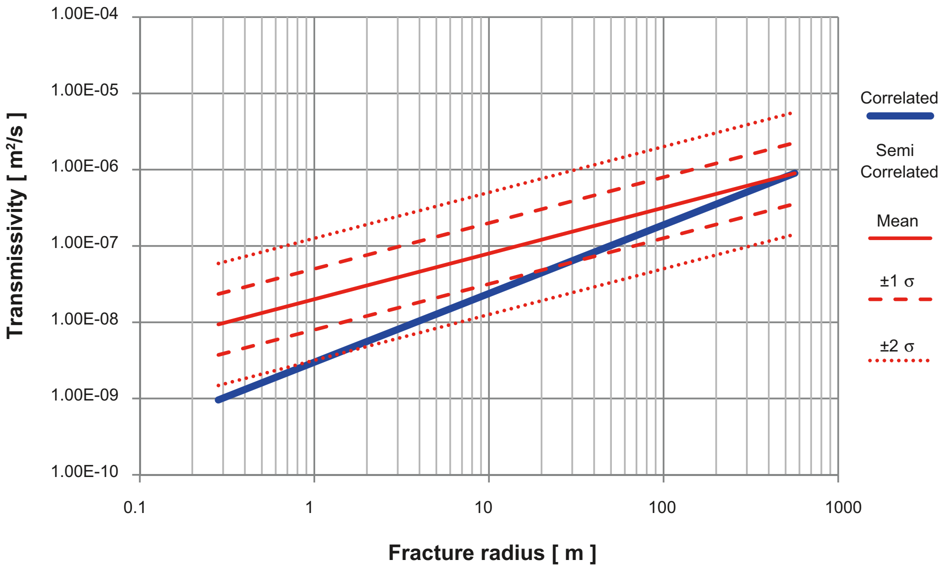

Figure 2-6. Relationship between radius and transmissivity for the WNW fracture set in HRD_C at depth 400 to 650 m below sea level.

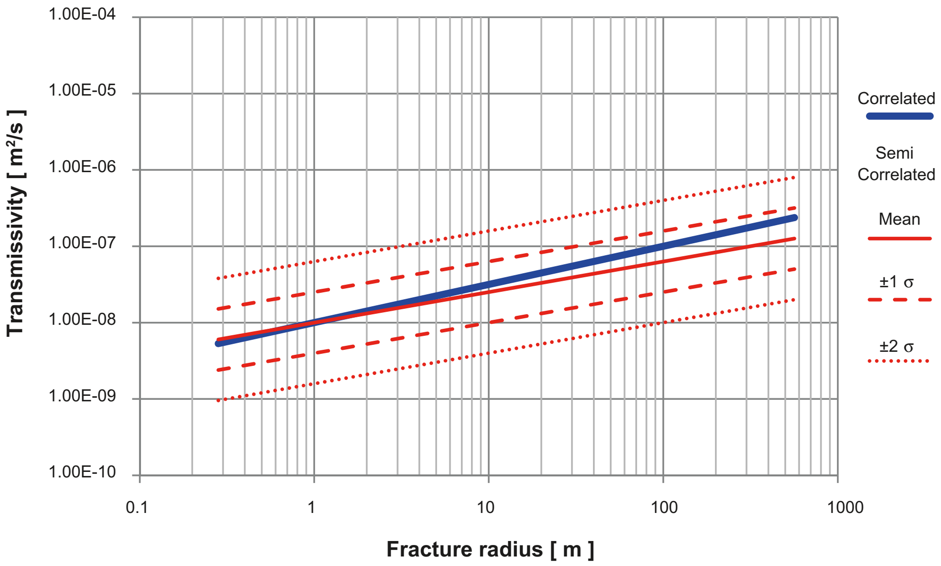

Figure 2-7. *Relationship between radius and transmissivity for the N-S striking fracture set in HRD_C at depth 400 to 650 m below sea level.*

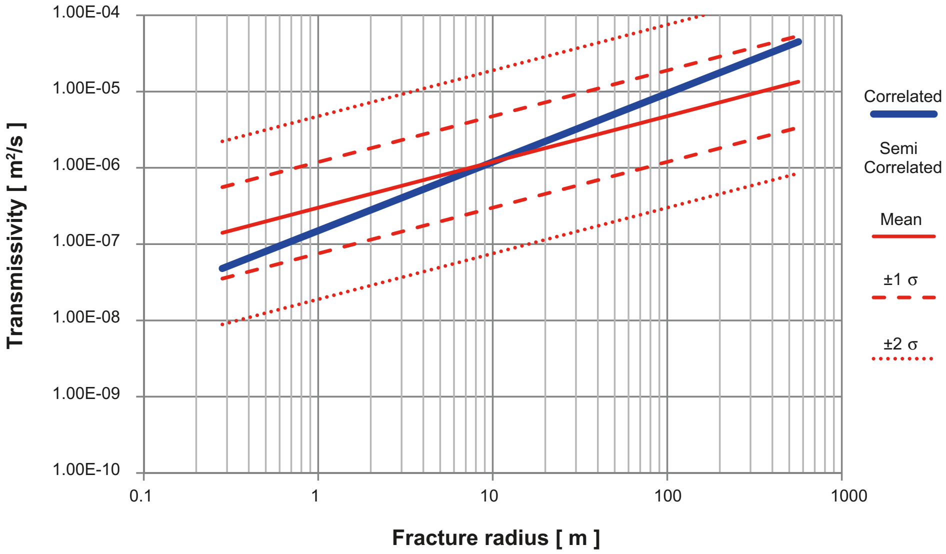

Figure 2-8. *Relationship between radius and transmissivity for the sub horizontal fracture set in HRD_C at depth 400 to 650 m below sea level.*

Table 2-6. Parameters for the relationship between transmissivity and radius in HRD_W at depth 400 to 650 m below sea level. Excerption from Table 10-12 in /Rhén et al. 2008/.

Set	Correlated		Semi correlated		
	a	b	a	b	σ
ENE	$1 \cdot 10^{-9}$	0.6	$3 \cdot 10^{-9}$	0.6	0.5
WNW	$2 \cdot 10^{-8}$	0.7	$3 \cdot 10^{-8}$	0.6	0.5
N-S	$1 \cdot 10^{-8}$	0.6	$3 \cdot 10^{-8}$	0.4	0.4
SubH	$1.2 \cdot 10^{-7}$	1.2	$5 \cdot 10^{-7}$	0.4	1.0

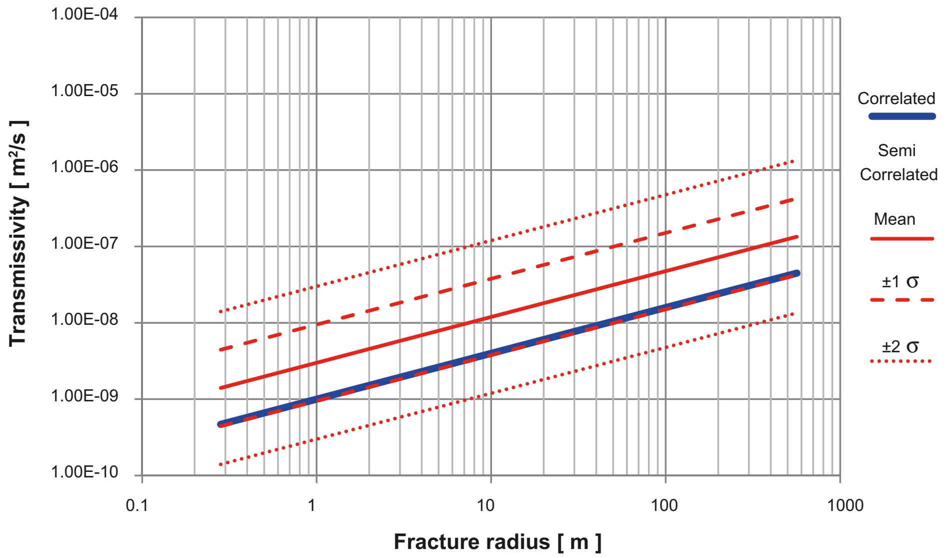

__Figure 2-9.__ Relationship between radius and transmissivity for the ENE striking fracture set in HRD_W at depth 400 to 650 m below sea level.

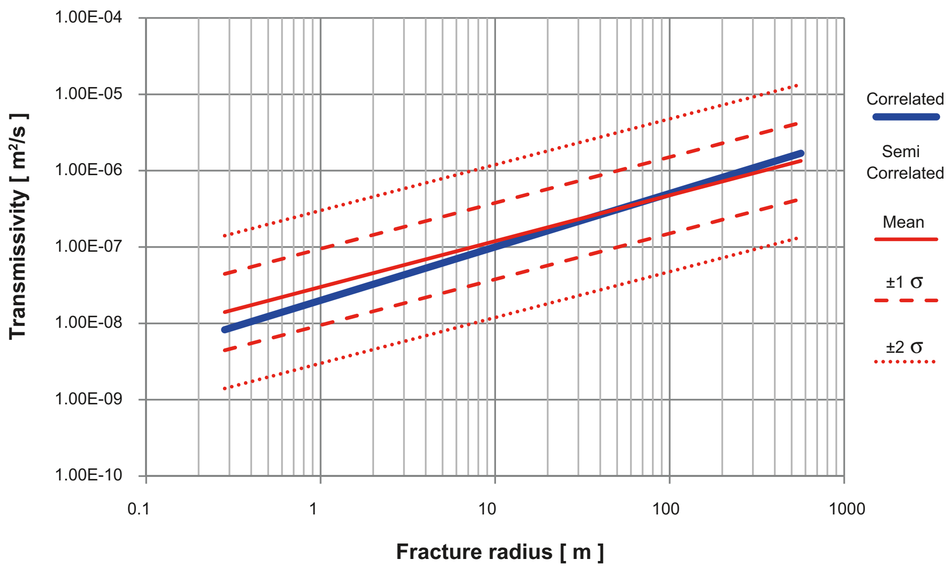

__Figure 2-10.__ Relationship between radius and transmissivity for the WNW striking fracture set in HRD_W at depth 400 to 650 m below sea level.

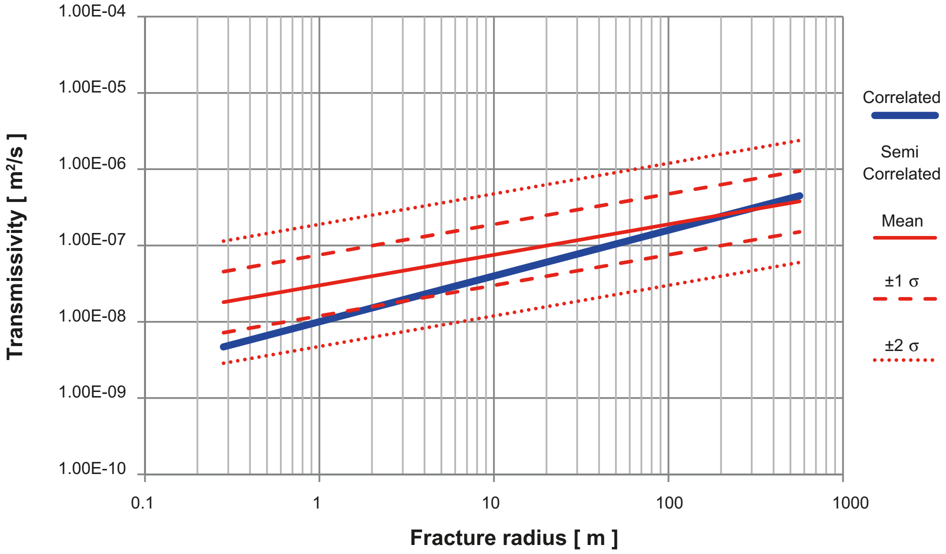

Figure 2-11. *Relationship between radius and transmissivity for the N-S striking fracture set in HRD_W at depth 400 to 650 m below sea level.*

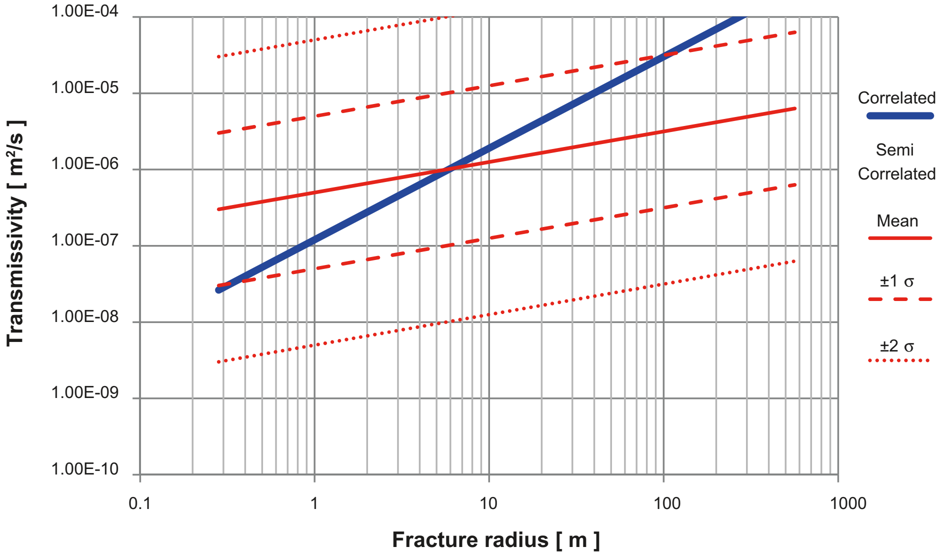

Figure 2-12. *Relationship between radius and transmissivity for the sub horizontal fracture set in HRD_W at depth 400 to 650 m below sea level.*

Table 2-7. Parameters for the relationship between transmissivity and radius in HRD_EW007 at depth 400 to 650 m below sea level. Excerption from Table 10-15 in /Rhén et al. 2008/.

Set	Correlated		Semi correlated		
	a	b	a	b	σ
ENE	$3 \cdot 10^{-8}$	0.4	$3 \cdot 10^{-8}$	0.4	0.4
WNW	$1 \cdot 10^{-7}$	0.3	$1 \cdot 10^{-7}$	0.3	0.3
N-S	$3 \cdot 10^{-7}$	0.4	$3 \cdot 10^{-7}$	0.4	0.4
SubH	$3 \cdot 10^{-8}$	0.6	$3 \cdot 10^{-8}$	0.6	0.6

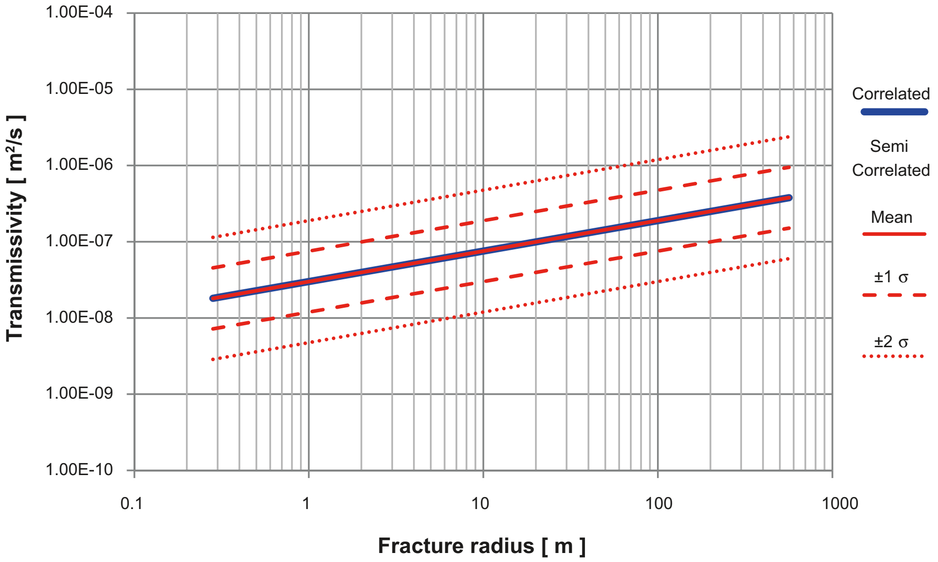

Figure 2-13. Relationship between radius and transmissivity for the ENE striking fracture set in HRD_EW007 at depth 400 to 650 m below sea level.

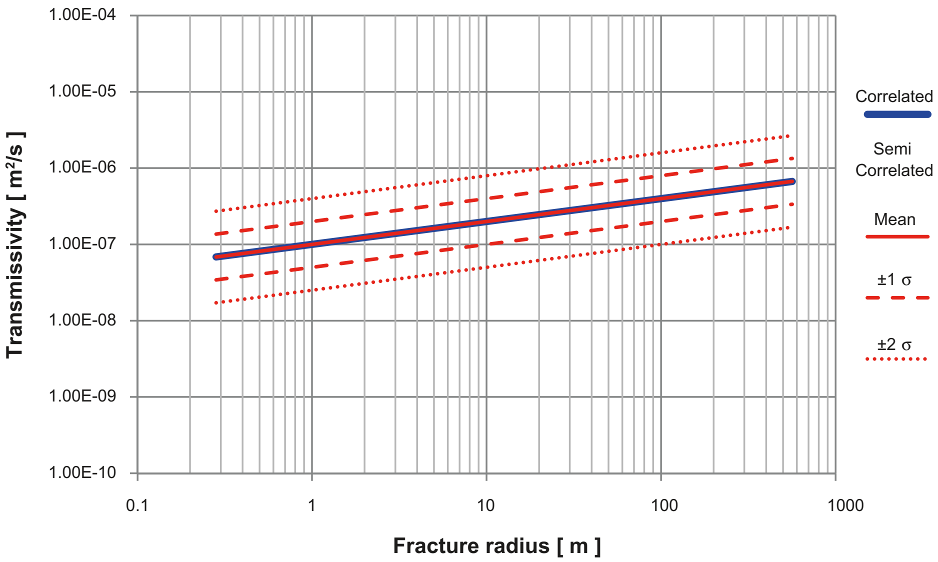

Figure 2-14. Relationship between radius and transmissivity for the WNW striking fracture set in HRD_EW007 at depth 400 to 650 m below sea level.

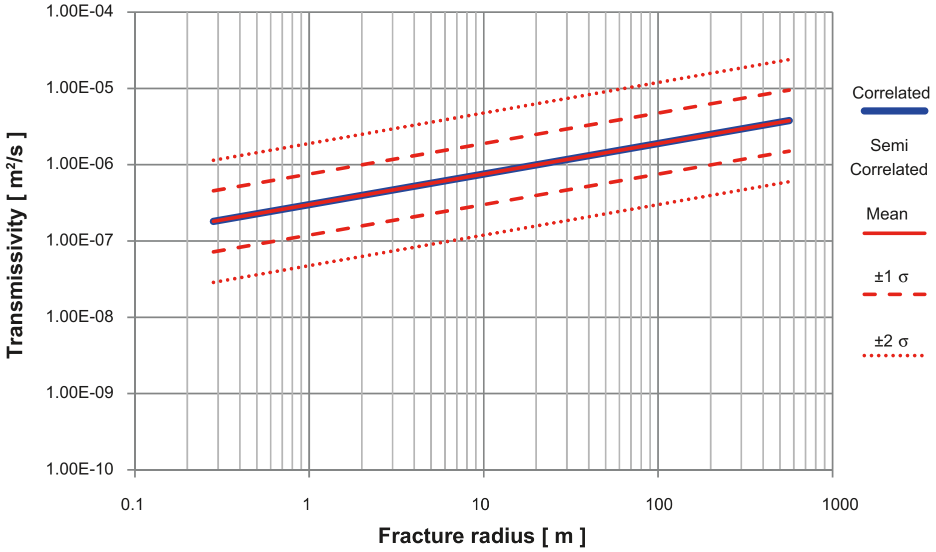

Figure 2-15. *Relationship between radius and transmissivity for the N-S striking fracture set in HRD_EW007 at depth 400 to 650 m below sea level.*

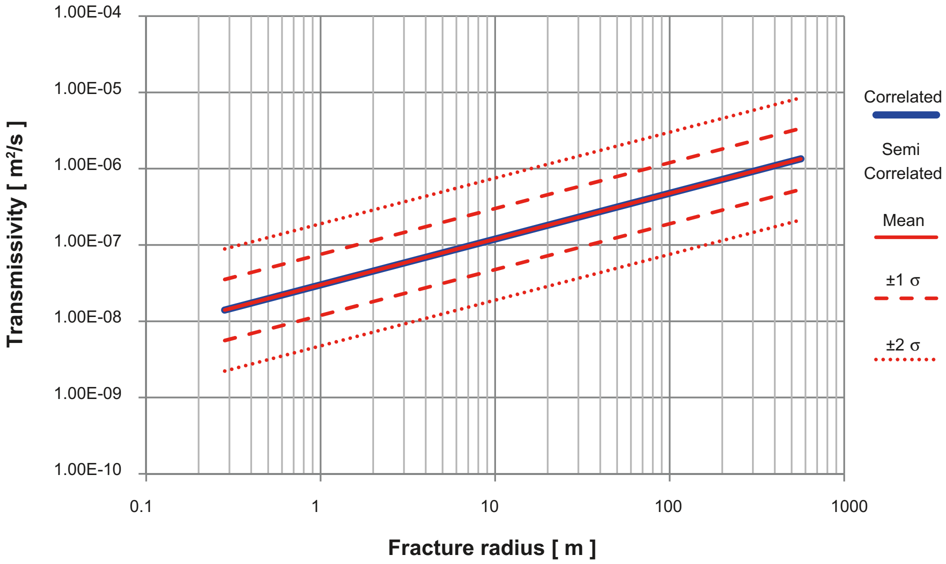

Figure 2-16. *Relationship between radius and transmissivity for the sub horizontal fracture set in HRD_EW007 at depth 400 to 650 m below sea level.*

2.4.2 Forsmark

All fractures within the same FFM and depth interval follow the same relationship between radius and transmissivity in Forsmark /Follin et al. 2007b/. An extraction of the transmissivity models, displayed in Table 11-18 in /Follin et al. 2007b/, are shown as Table 2-8. A visualisation of the relationship between radius and transmissivity is shown in Figure 2-17. Though observe that the y-axis is shifted 2 orders of magnitude compared to the graphs in section 2.4.1.

Table 2-8. Parameters for the relationship between transmissivity and radius in FFM01 at depth beneath 400 m below sea level. Excerption from Table 3-7 in /Follin et al. 2007a/ and 11-20 in /Follin et al. 2007b/.

Set	Correlated		Semi correlated		
	a	b	a	b	σ
All	$1.8 \cdot 10^{-10}$	1.0	$5.3 \cdot 10^{-11}$	0.5	1.0

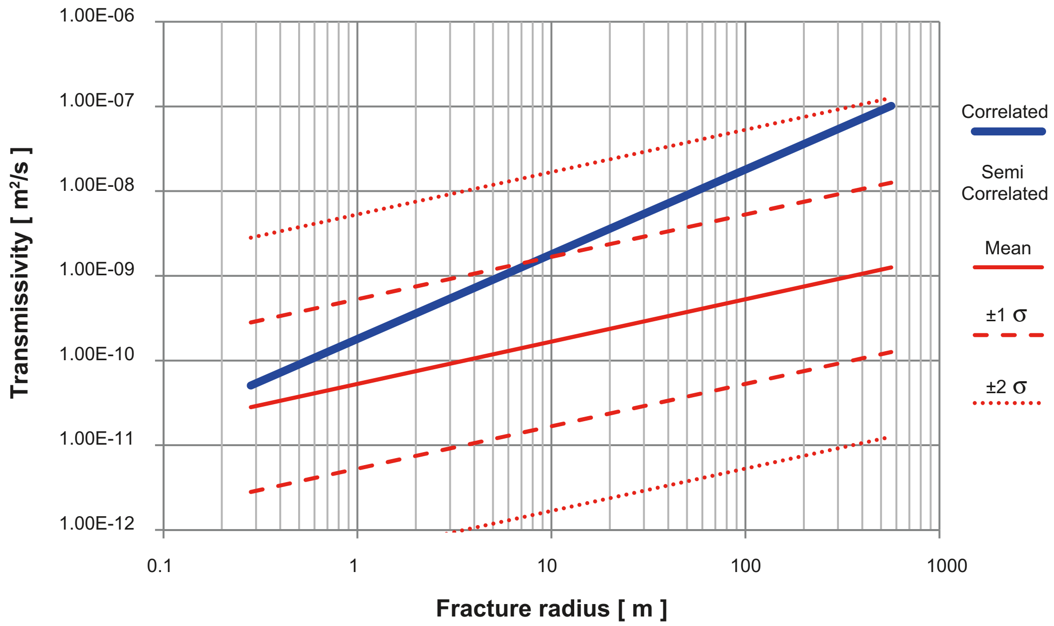

Figure 2-17. Relationship between radius and transmissivity for all fracture sets in FFM01 at depth beneath 400 m below sea level.

2.4.3 Transformation to DarcyTools parameters

The assignment of transmissivity in DarcyTools is done using other parameters than those stated in Table 2-5 to Table 2-8, see Formula 2-10.

$$T = \min\left[a_T \cdot 10^{d_T \cdot U} \cdot (l/100)^{b_T}, c_T\right]$$ Formula 2-10

Where

T = transmissivity [m^2/s]

a_T = location parameter [m^2/s]

b_T = shape parameter [–]

c_T = maximum allowed transmissivity [m^2/s]

d_T = factor for scaling the uncertainty [–]

U = Stochastic number from uniform distribution in the interval –0.5 to 0.5 [–]

l = side length of a square fracture [m]

The relationship between the parameters given inTable 2-5 to Table 2-8 and the ones used in DarcyTools for the correlated transmissivity model can be written as

$$b_T = b$$ Formula 2-11

$$a_T = a \cdot \left(100/\sqrt{\pi}\right)^b$$ Formula 2-12

Unfortunately DarcyTools uses a rectangular distribution instead of a normal distribution when calculating the stochastic part of the transmissivity. Hence the assignment of transmissivity according to the semi correlated model is performed in the evaluation program using Formula 2-9, see Appendix A2.

2.5 Sampling model

Though the planned deposition tunnels will have a cross section area of 19 m^2 and will be 300 m long /SKB 2007/ the sampling model is kept simple and is represented by a 100 m long line, i.e. infinitesimal radius. Also the deposition holes have a sample model that is 100 m long and infinitesimal radius, though they are 8 m deep and have a diameter of 1.75 m /SKB 2007/. The statistics are based on 100 m and 20 m sections for the horizontal lines and 8 m sections for the vertical. Since the spatial model of the fractures is poissonian it is possible to put all the sampling lines after each other and then divide them into suitable pieces without violating the statistics.

2.6 Nested volumes

The nature of powerlaw distributions is that it produces lots of small values and fewer large ones, especially for distributions with large shape parameter. Small fractures are less important for the connectivity over large distances than the larger ones and hence only contribute to connectivity in vicinity of larger structures. Therefore it is more efficient to divide the volume where fractures are generated into nested boxes so that the smaller volume the smaller fractures are generated.

However the volumes can not be too small as a fracture might reach from the boundary to the sampling line. In the presented work there are four boxes each a little bit larger than the diagonal of the largest square fracture generated. The sampling line is 100 m and hence the extensions of the boxes are 100 m longer in the direction of the sampling line than perpendicular to the line. The extension of the boxes and the fractures generated within them is tabulated in Table 2-9.

Consequently the boundaries do not reflect any natural boundaries at Laxemar or Forsmark, but only chosen large enough not to make it possible for any single fracture to extend from the boundary to the sampling line.

2.7 Rotation of model

To decrease the computational work as much as possible for the horizontal sampling line, the x-axes of the boxes are made parallel to the orientation of the deposition tunnels, i.e. rotated 40° CW (130°–90°) for the Laxemar model respectively 50° CW (140°–90°) for the Forsmark model. The fracture mean poles are rotated in the same way to get the right statistics. For deeper insight see Appendix A3. No rotation is needed when modelling sampling in the vertical direction.

Table 2-9 The extension of the four boxes and the corresponding fracture sizes generated in them.

Box extension Along sampling line	Perpendicular to sampling line	Fracture radii generated Smallest	Largest
3,100	3,000	56.4	564
400	300	5.64	56.4
130	30	0.56	5.64
103	3	0.28	0.56

2.8 Summary of input to DarcyTools

Table 2-10 to Table 2-17 summarize the parameters that are used for fracture generation and assignment of correlated transmissivities in the different HRDs for horizontal (rotated) and vertical sampling. The construction of all indata files used in the project is provided in Appendix A3 as Visual Basic .Net 2003 code.

Table 2-10. Used DarcyTools fracture parameters for horizontal sampling in HRD_C at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	8.67816	–4.06515	–0.56934	–2.8	0.03513664	5.04799E–07	0.7
WNW	3.28623	–11.53640	–0.33506	–2.5	0.09681898	1.13085E–07	0.9
N-S	–5.92832	–4.93929	–1.13945	–2.9	0.03731982	7.51126E–08	0.5
SubH	0.12163	1.10175	–11.94870	–2.9	0.04605424	5.65423E–06	0.9

Table 2-11. Used DarcyTools fracture parameters for horizontal sampling in HRD_W at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	–12.98753	7.49836	–0.31414	–2.8	0.01571902	1.12423E–08	0.6
WNW	2.09694	–10.68821	–0.41843	–2.55	0.04136992	3.36533E–07	0.7
N-S	–8.95992	–6.80095	–2.39098	–2.55	0.03760902	1.12423E–07	0.6
SubH	–0.92564	–0.59882	–11.04512	–2.65	0.04285236	1.51667E–05	1.2

Table 2-12. Used DarcyTools fracture parameters for horizontal sampling in HRD_EW007 at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	8.99138	–5.79455	–0.26142	–2.95	0.05053132	1.50554E–07	0.4
WNW	–4.16160	15.86309	–0.05725	–2.5	0.18709610	3.35295E–07	0.3
N-S	6.61600	5.77151	–0.59853	–2.95	0.04686963	1.50554E–06	0.4
SubH	1.45035	–0.22193	–9.58839	–2.95	0.06737509	3.37269E–07	0.6

Table 2-13. Used DarcyTools fracture parameters for vertical sampling in HRD_C at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	4.03483	–8.69230	–0.56934	–2.8	0.03513664	5.04799E–07	0.7
WNW	–4.89806	–10.94974	–0.33506	–2.5	0.09681898	1.13085E–07	0.9
N-S	7.71628	0.02693	–1.13945	–2.9	0.03731982	7.51126E–08	0.5
SubH	0.80137	0.76581	–11.94870	–2.9	0.04605424	5.65423E–06	0.9

Table 2-14. Used DarcyTools fracture parameters for vertical sampling in HRD_W at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	–5.12918	14.09230	–0.31414	–2.8	0.01571902	1.12423E–08	0.6
WNW	–5.26390	–9.53553	–0.41843	–2.55	0.04136992	3.36533E–07	0.7
N-S	–11.23527	0.54950	–2.39098	–2.55	0.03760902	1.12423E–07	0.6
SubH	–1.09400	0.13626	–11.04512	–2.65	0.04285236	1.51667E–05	1.2

Table 2-15. Used DarcyTools fracture parameters for vertical sampling in HRD_EW007 at depth 400 to 650 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
ENE	3.16313	–10.21843	–0.26142	–2.95	0.05053132	1.50554E–07	0.4
WNW	7.00863	14.82686	–0.05725	–2.5	0.18709610	3.35295E–07	0.3
N-S	8.77800	0.16855	–0.59853	–2.95	0.04686963	1.50554E–06	0.4
SubH	0.96837	–1.10228	–9.58839	–2.95	0.06737509	3.37269E–07	0.6

Table 2-16. Used DarcyTools fracture parameters for horizontal sampling in FFM01 at depth below 400 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
NS	–15.71407	–8.35532	–0.31065	–2.5	0.01229863	1.01554E–08	1.0
NE	–14.21300	1.49385	–0.49906	–2.75	0.01617487	1.01554E–08	1.0
NW	2.22779	12.63443	–1.34842	–3.1	0.00554401	1.01554E–08	1.0
EW	–8.02518	11.46114	–0.48859	–3.1	0.00220629	1.01554E–08	1.0
HZ	–0.74974	0.74974	–15.16297	–2.38	0.01973405	1.01554E–08	1.0

Table 2-17. Used DarcyTools fracture parameters for vertical sampling in FFM01 at depth below 400 m below sea level.

Set	λ_1 (–)	λ_2 (–)	λ_3 (–)	D (–)	I (m^{-3})	a_T (m^2/s)	b_T (–)
NS	–16.50136	6.66698	–0.31065	–2.5	0.01229863	1.01554E–08	1.0
NE	–7.99159	11.84802	–0.49906	–2.75	0.01617487	1.01554E–08	1.0
NW	11.11053	6.41467	–1.34842	–3.1	0.00554401	1.01554E–08	1.0
EW	3.62126	13.51472	–0.48859	–3.1	0.00220629	1.01554E–08	1.0
HZ	0.09241	1.05626	–15.16297	–2.38	0.01973405	1.01554E–08	1.0

3 Results

The results in this study rely on 1,000 realisations for each HRD/FFM and sampling direction. Each sampling section is 100 m and hence the statistics are calculated from 100,000 m sampling line. This should be enough for reliable average and standard deviations, but one should not draw any major conclusions regarding the tails of the statistics. If the tails are of interest sensitivity analysis has to be carried out.

Throughout this section the statistics are first presented in a graph and underneath a table of the key values is given of the same graph to facilitate extracting of accurate numbers.

3.1 Distances between features intersecting horizontal scan line

By putting all the 1,000 realisations of 100 m sections after another an assembled 100 km long scanline is obtained. From this assembled scanline the distances between the features remaining in the model, when truncating data at different transmissivity values, are calculated. Since only distances *between* features are calculated there will be one distance less than the number of fractures intersecting the scanline to calculate the statistics from, i.e. neither the distance between the start of the 100 km scan line and the first fracture nor the distance between the last fracture and the end of the assembled scanline is part of the statistics. Consequently this approach implies that there must be more than one feature intersecting the assembled scan line to obtain any value.

The transmissivity limits of which features that are preserved in or discarded from the model when calculating the distances, are chosen to be $4\cdot10^{-9}$, $2\cdot10^{-7}$, $5\cdot10^{-7}$, $1\cdot10^{-6}$ m^2/s based on penetrability of different grouting materials. An additional limit $3\cdot10^{-8}$ m^2/s is introduced to minimise the gap between the two lower limits. In the Figures 3-1 to 3-8 the blue line is representing all features intersecting the scanline, the green line only contain features with transmissivity larger than $4\cdot10^{-9}$ m^2/s and so on to the purple line that only contains the relatively few features with transmissivity larger than $1\cdot10^{-6}$ m^2/s.

3.1.1 Laxemar

The graphs in Figure 3-1 to Figure 3-6 are shown as Complementary Cumulative Density Functions, CCDF, i.e. the probability that a distance between two features is larger than the specific distance at the different HRDs of Laxemar. Below each graph there is a table showing the different percentiles for the Cumulative Density Function, CDF, i.e. the probability that the distance is smaller than the, in the table, given value.

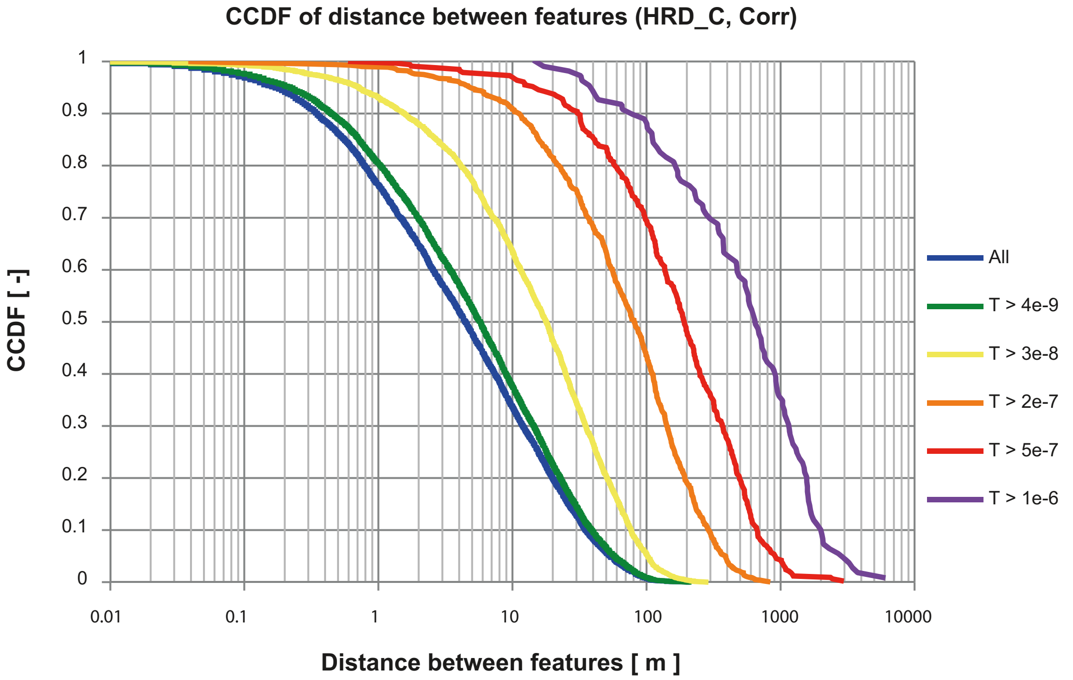

Figure 3-1. The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_C with the correlated transmissivity model

Table 3-1. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_C with the correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.17	0.35	1.09	4.45	15.61	35.17	53.22	12.60	19.76
$T>4\cdot10^{-9}$	0.22	0.45	1.44	5.75	17.92	39.03	57.36	14.14	20.93
$T>3\cdot10^{-8}$	0.72	1.56	5.54	17.85	41.91	76.75	101.81	30.12	35.00
$T>2\cdot10^{-7}$	4.85	11.21	30.30	81.04	162.10	288.19	383.89	119.11	123.95
$T>5\cdot10^{-7}$	14.60	30.77	76.20	190.83	431.52	663.05	899.15	297.84	341.48
$T>1\cdot10^{-6}$	38.18	75.72	227.03	639.94	1,301.86	1,905.15	2,545.07	897.27	923.19

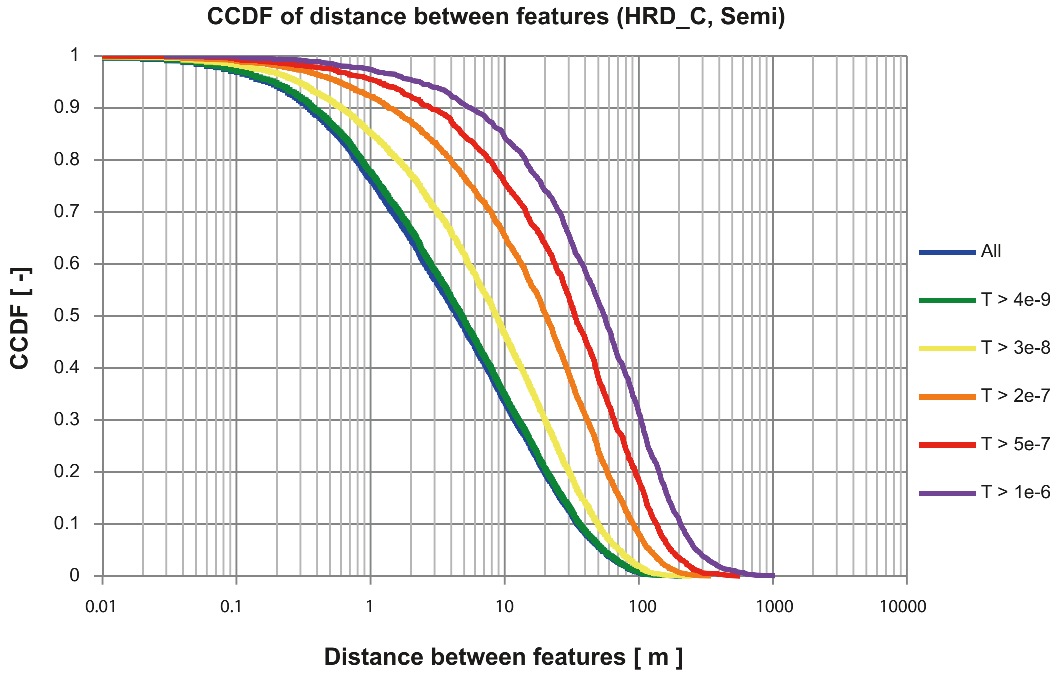

Figure 3-2. The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_C with the semi correlated transmissivity model

Table 3-2. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_C with the semi correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.17	0.35	1.09	4.45	15.61	35.17	53.22	12.60	19.76
$T>4\cdot10^{-9}$	0.19	0.37	1.19	4.80	16.36	36.32	54.40	13.09	20.17
$T>3\cdot10^{-8}$	0.29	0.61	2.34	8.67	24.60	49.79	71.65	18.48	25.05
$T>2\cdot10^{-7}$	0.59	1.37	5.60	20.46	49.08	90.70	120.88	35.07	41.49
$T>5\cdot10^{-7}$	1.16	2.81	10.40	33.51	78.81	135.63	174.04	55.14	62.58
$T>1\cdot10^{-6}$	2.35	5.61	19.02	55.51	116.80	204.88	268.67	87.35	102.00

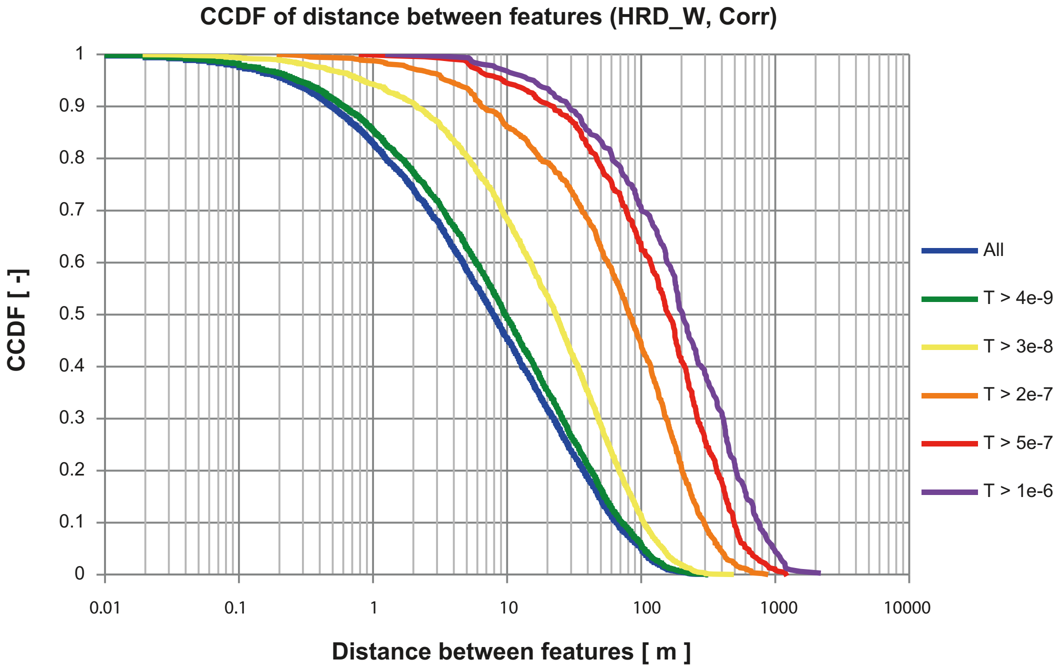

Figure 3-3. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_W with the correlated transmissivity model*

Table 3-3. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_W with the correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.23	0.49	1.89	7.89	28.41	65.55	99.83	22.74	34.83
$T>4\cdot10^{-9}$	0.28	0.61	2.41	9.67	33.00	72.35	104.72	25.54	37.67
$T>3\cdot10^{-8}$	0.84	2.11	7.09	23.17	57.15	104.74	145.98	41.37	50.41
$T>2\cdot10^{-7}$	3.54	6.46	28.06	81.30	178.28	294.03	385.18	122.99	128.77
$T>5\cdot10^{-7}$	9.41	21.90	60.72	159.02	305.92	492.70	597.94	217.62	206.98
$T>1\cdot10^{-6}$	15.71	28.73	82.80	203.68	442.35	744.55	952.39	312.44	310.84

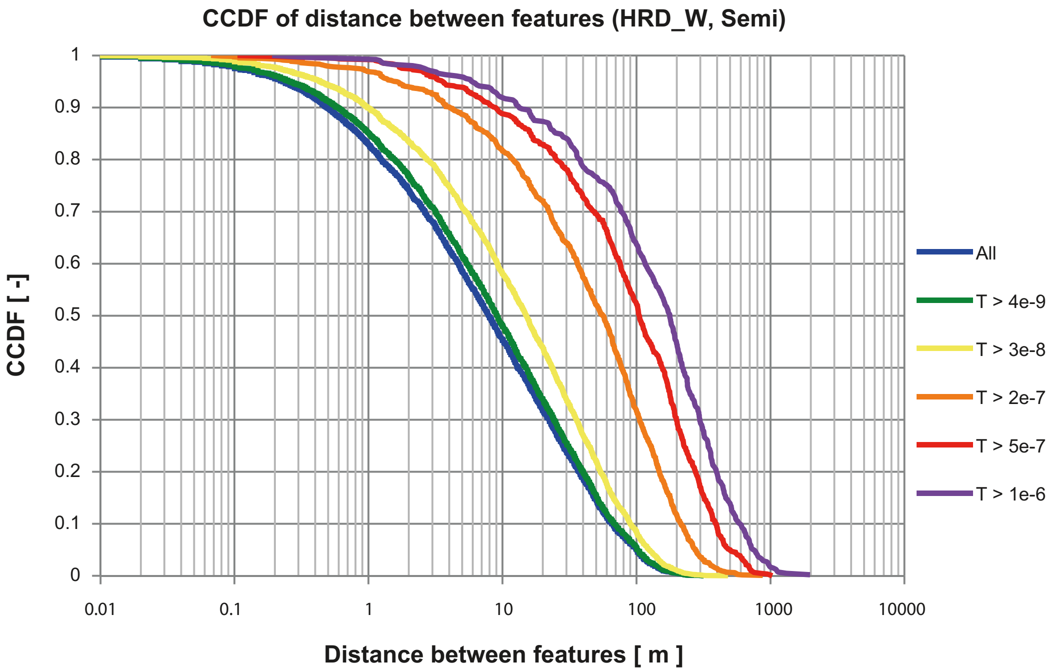

Figure 3-4. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_W with the semi correlated transmissivity model*

Table 3-4. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_W with the semi correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.23	0.49	1.89	7.89	28.41	65.55	99.83	22.74	34.83
$T>4\cdot10^{-9}$	0.26	0.59	2.25	9.11	30.85	68.04	101.76	24.24	35.97
$T>3\cdot10^{-8}$	0.44	0.99	3.95	15.03	43.99	89.93	124.23	32.54	43.85
$T>2\cdot10^{-7}$	1.60	4.05	16.13	56.27	126.96	214.12	275.10	86.75	96.30
$T>5\cdot10^{-7}$	3.59	8.83	35.75	104.51	225.09	387.53	484.97	161.20	166.88
$T>1\cdot10^{-6}$	5.90	13.26	60.57	177.18	342.50	576.20	725.87	242.33	251.81

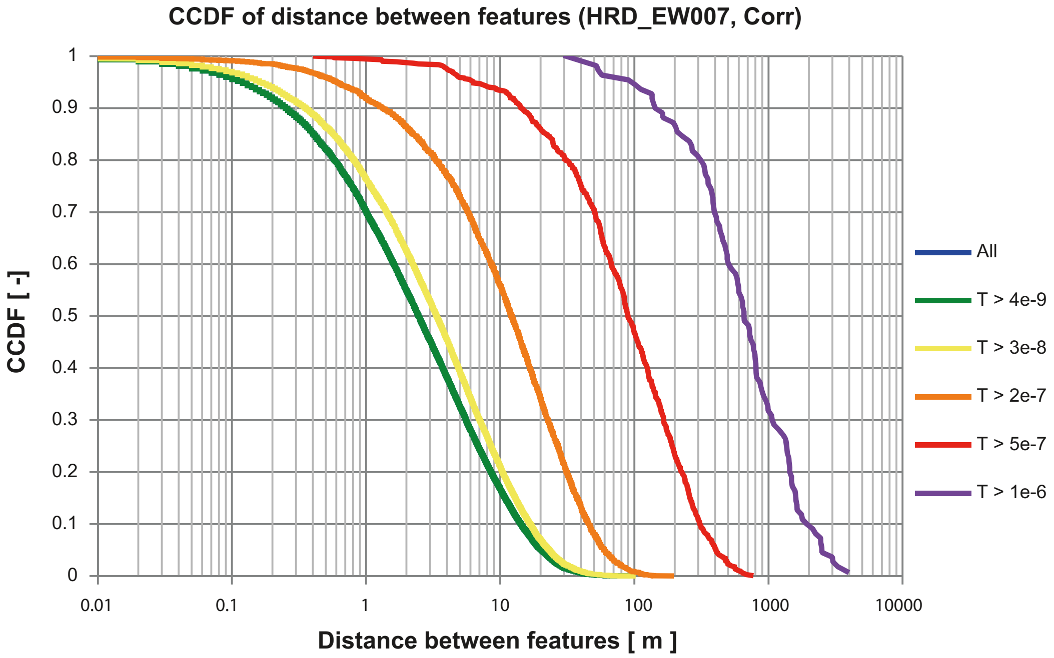

Figure 3-5. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_EW007 with the correlated transmissivity model*

Table 3-5. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_EW007 with the correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.12	0.25	0.79	2.47	6.85	14.29	20.28	5.33	7.33
$T>4\cdot10^{-9}$	0.12	0.25	0.79	2.47	6.85	14.29	20.28	5.33	7.33
$T>3\cdot10^{-8}$	0.16	0.35	1.08	3.36	8.54	16.76	23.23	6.44	8.31
$T>2\cdot10^{-7}$	0.60	1.34	4.55	12.19	26.94	45.56	59.55	18.97	20.45
$T>5\cdot10^{-7}$	6.29	14.31	40.02	89.54	191.22	306.88	404.88	135.00	130.55
$T>1\cdot10^{-6}$	95.11	142.58	359.35	657.04	1,352.59	1,802.94	2,479.28	892.06	767.66

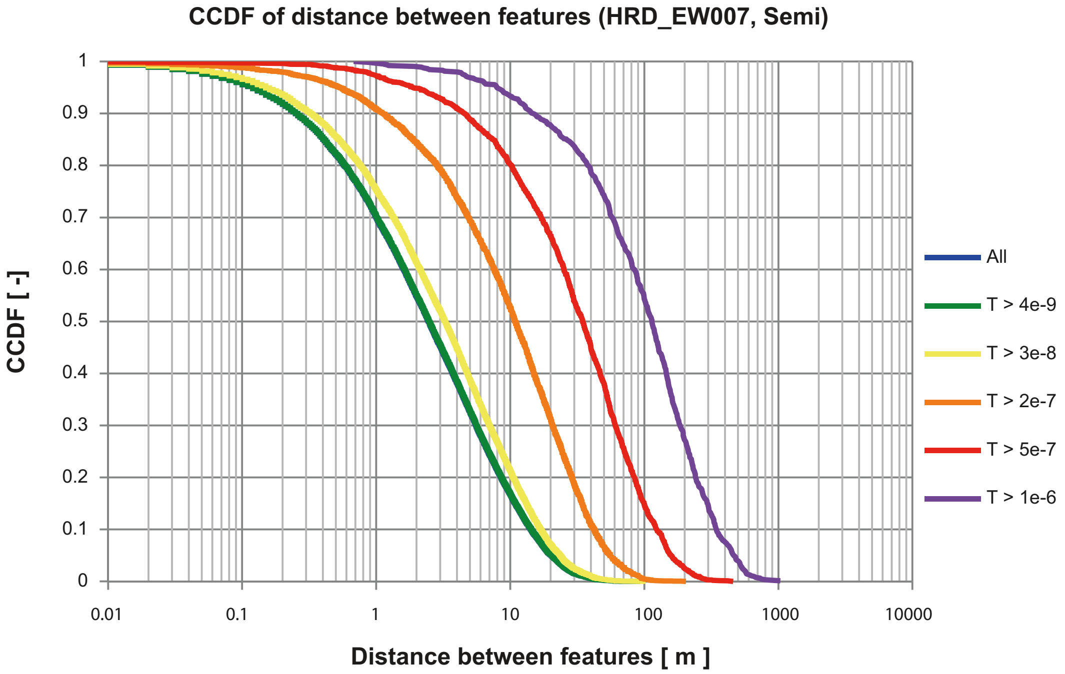

Figure 3-6. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_EW007 with the semi correlated transmissivity model*

Table 3-6. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_EW007 with the semi correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.12	0.25	0.79	2.47	6.85	14.29	20.28	5.33	7.33
$T>4\cdot10^{-9}$	0.12	0.26	0.79	2.49	6.89	14.39	20.43	5.36	7.37
$T>3\cdot10^{-8}$	0.15	0.33	1.02	3.27	8.60	16.89	23.55	6.45	8.48
$T>2\cdot10^{-7}$	0.54	1.13	3.79	10.93	24.33	42.26	55.65	17.38	19.29
$T>5\cdot10^{-7}$	1.89	4.53	13.06	34.72	71.12	123.20	157.36	51.63	53.79
$T>1\cdot10^{-6}$	7.94	14.71	48.30	114.08	212.68	343.06	459.30	153.40	144.87

3.1.2 Forsmark

The graphs in Figure 3-7 to Figure 3-8 are shown as Complementary Cumulative Density Functions, CCDF, of the distance for FFM01 at Forsmark. The scales on the axes are kept the same as in section 3.1.1 to facilitate comparison between the two sites. However, this results in the CCDF of the distance between features with transmissivity larger than $2\cdot10^{-7}$ m^2/s being entirely outside the plot area in Figure 3-8. The CDF data are nevertheless shown in Table 3-8 and thus the graph can be re-created by the interested reader.

Also noticeable is that there are less than two features with transmissivity larger than $2\cdot10^{-7}$ m^2/s for the correlated transmissivity model and less than two features with transmissivity larger than $5\cdot10^{-7}$ m^2/s for the semi correlated transmissivity model intersecting the assembled 100 km scanline and accordingly there will not be any graphs or data shown in Figure 3-7, Figure 3-8, Table 3-7, or Table 3-8 for these cases.

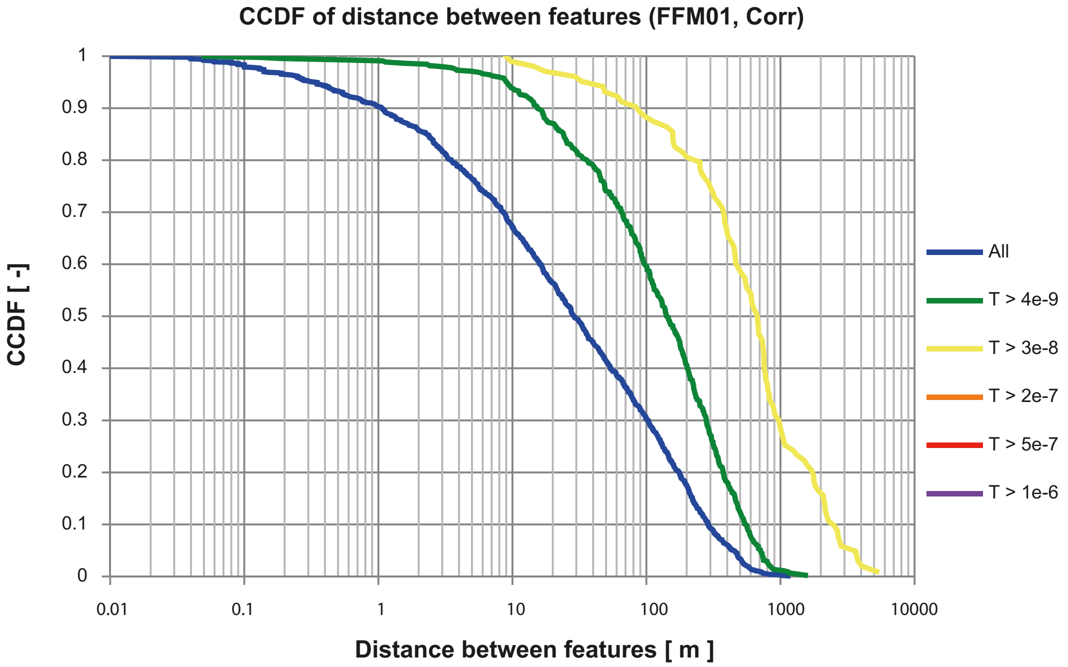

Figure 3-7. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for FFM01 with the correlated transmissivity model*

Table 3-7. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for FFM01 with the correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.34	1.07	5.59	28.75	131.60	288.16	433.46	98.54	150.33
$T>4\cdot10^{-9}$	9.06	15.36	48.85	141.15	317.04	541.25	699.40	221.33	231.35
$T>3\cdot10^{-8}$	34.30	84.31	295.50	662.56	1,067.42	2,267.66	2,831.92	955.10	1,013.39
$T>2\cdot10^{-7}$	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$T>5\cdot10^{-7}$	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$T>1\cdot10^{-6}$	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

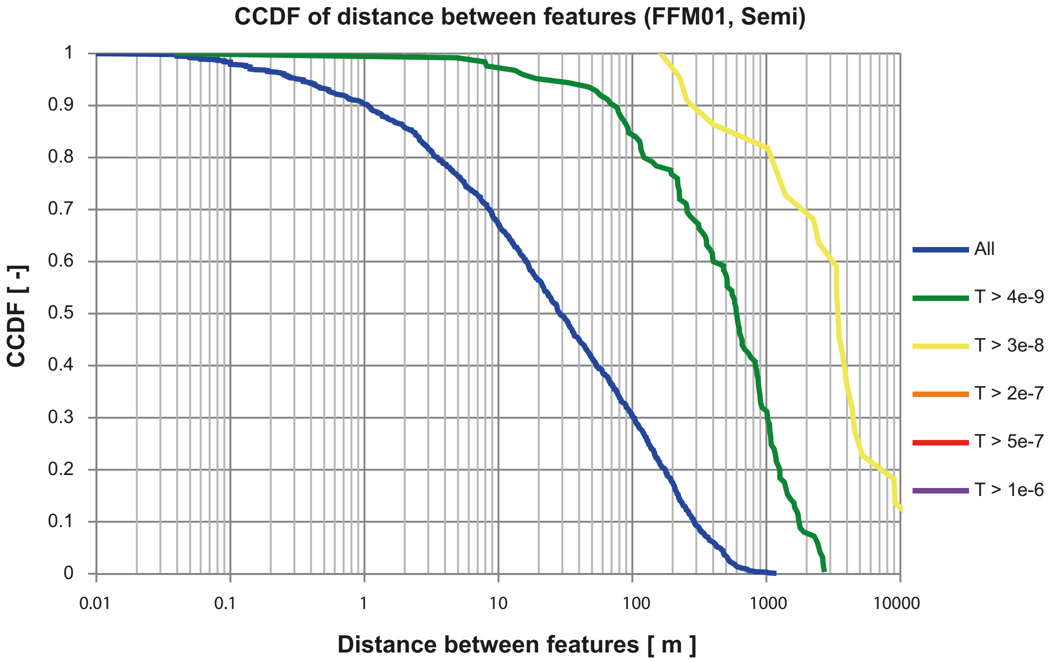

Figure 3-8. *The complementary cumulative density functions of distances between fractures with transmissivity larger than the given values in the legend. Data shown for HRD_EW007 with the semi correlated transmissivity model*

Table 3-8. The percentiles for the cumulative density function of distances between fractures with transmissivity larger than the given values to the left in the table. Data shown for HRD_EW007 with the semi correlated transmissivity model.

Trunc.	5%	10%	25%	50%	75%	90%	95%	average	std dev
all	0.34	1.07	5.59	28.75	131.60	288.16	433.46	98.54	150.33
$T>4\cdot10^{-9}$	21.97	72.29	217.73	595.80	1,080.98	1,737.79	2,421.38	764.84	695.31
$T>3\cdot10^{-8}$	226.61	270.28	1,243.96	3,382.49	4,532.49	9,124.92	13,972.89	4,193.17	4,167.47
$T>2\cdot10^{-7}$	16,650.3	18,077.4	22,358.7	29,494.2	36,629.6	40,910.9	42,338.0	29,494.2	20,182.1
$T>5\cdot10^{-7}$	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.
$T>1\cdot10^{-6}$	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.

3.2 Total transmissivity in tunnel sections

This chapter shows statistics of the sum of transmissivity of all features intersecting 20 m or 100 m horizontal sections. There are one thousand 100 m sections and five thousand 20 m sections that the statistics is calculated from. The data is shown as Complementary Cumulative Density Functions, CCDF, i.e. the probability that the sum of the transmissivity of the fractures in a section is larger than a specific value. The tables below each graph shows different percentiles for the Cumulative Density Function, CDF, i.e. the probability that the sum of transmissivities of intersecting fractures in a section is smaller than the, in the table, given value. Some sections are not intersected by any fracture, or the sum of transmissivities is below $1\cdot10^{-9}$ m^2/s and hence the graphs do not always start at 1.

To facilitate the reading of the CCDF graphs a histogram is provided where the relative number of sections in a specific transmissivity interval is shown. Below each histogram the values of the bars are presented in a table.

3.2.1 Laxemar

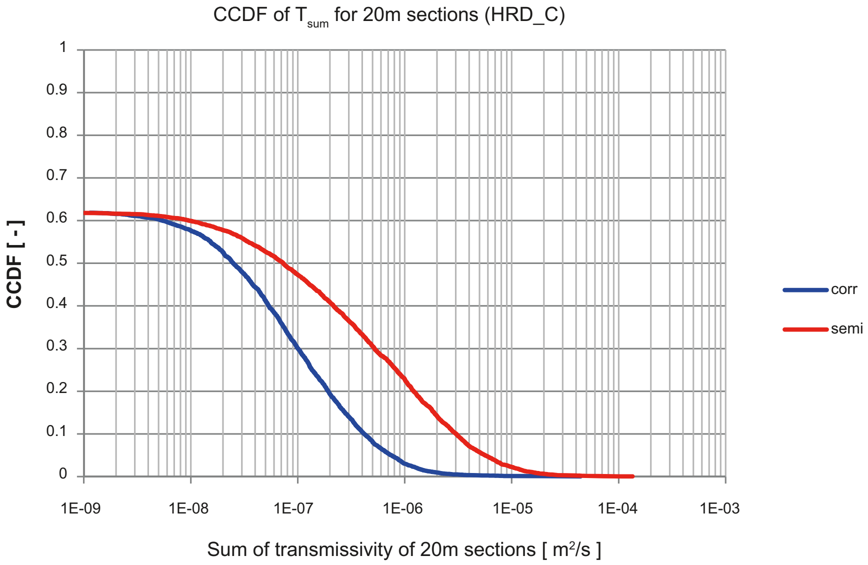

Figure 3-9. *The complementary cumulative density functions of sum of transmissivity in horizontal 20 m sections for the correlated and semi correlated transmissivity model for HRD_C at the depth interval 400 to 650 m below sea level.*

Table 3-9. The percentiles for the cumulative density function of sum of transmissivity in horizontal 20 m sections for HRD_C at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	2.49E–08	1.36E–07	4.16E–07	7.39E–07	1.86E–07	1.04E–06
Semi	–	–	–	7.32E–08	8.33E–07	3.03E–06	5.61E–06	1.19E–06	4.19E–06

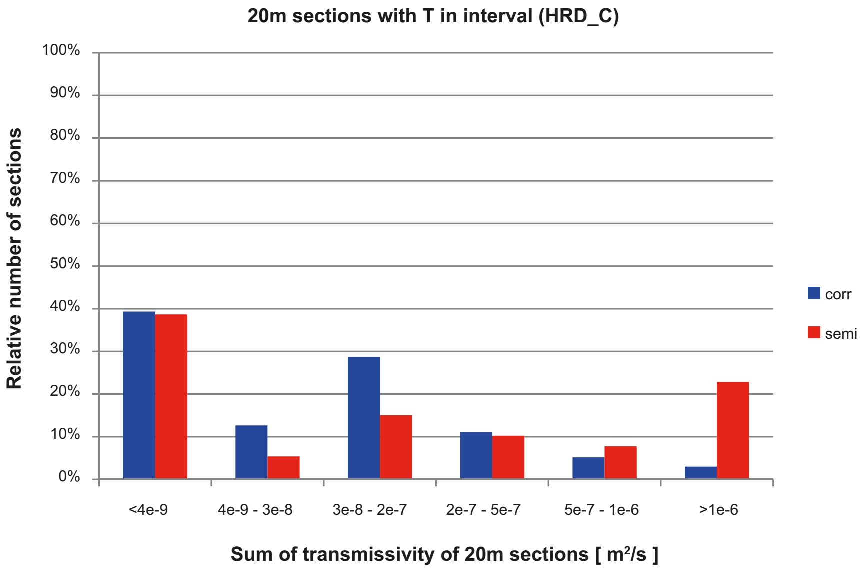

Figure 3-10. *Histogram of the graph in Figure 3-9, i.e. the relative number of 20 m sections with total transmissivity in the given intervals.*

Table 3-10. Tabulated numbers of Figure 3-10.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}–3\cdot10^{-8}$	$3\cdot10^{-8}–2\cdot10^{-7}$	$2\cdot10^{-7}–5\cdot10^{-7}$	$5\cdot10^{-7}–1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.3934	0.1266	0.2872	0.1112	0.0518	0.0298
semi	0.3868	0.0540	0.1506	0.1026	0.0776	0.2284

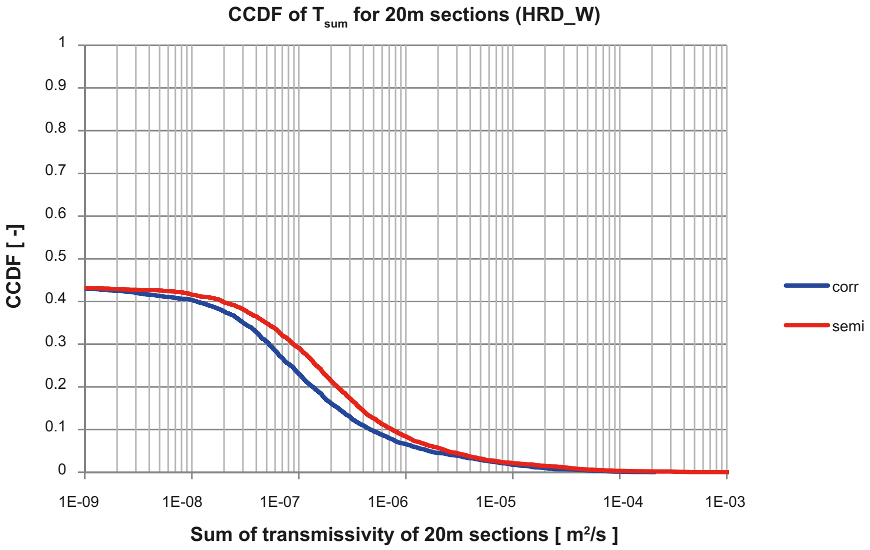

Figure 3-11. The complementary cumulative density functions of sum of transmissivity in horizontal 20 m sections for the correlated and semi correlated transmissivity model for HRD_W at the depth interval 400 to 650 m below sea level.

Table 3-11. The percentiles for the cumulative density function of sum of transmissivity in horizontal 20 m sections for HRD_W at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	8.33E–08	4.64E–07	1.62E–06	8.98E–07	7.11E–06
Semi	–	–	–	–	1.45E–07	7.46E–07	2.41E–06	2.20E–06	3.97E–05

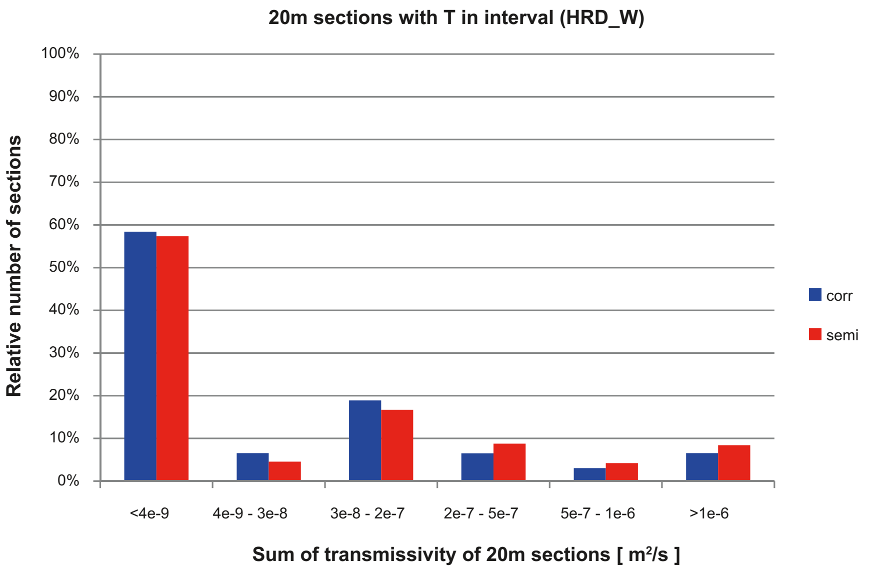

Figure 3-12. Histogram of the graph in Figure 3-11, i.e. the relative number of 20 m sections with total transmissivity in the given intervals.

Table 3-12. Tabulated numbers of Figure 3-12.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.5842	0.0656	0.1890	0.0650	0.0306	0.0656
semi	0.5734	0.0456	0.1672	0.0876	0.0422	0.0840

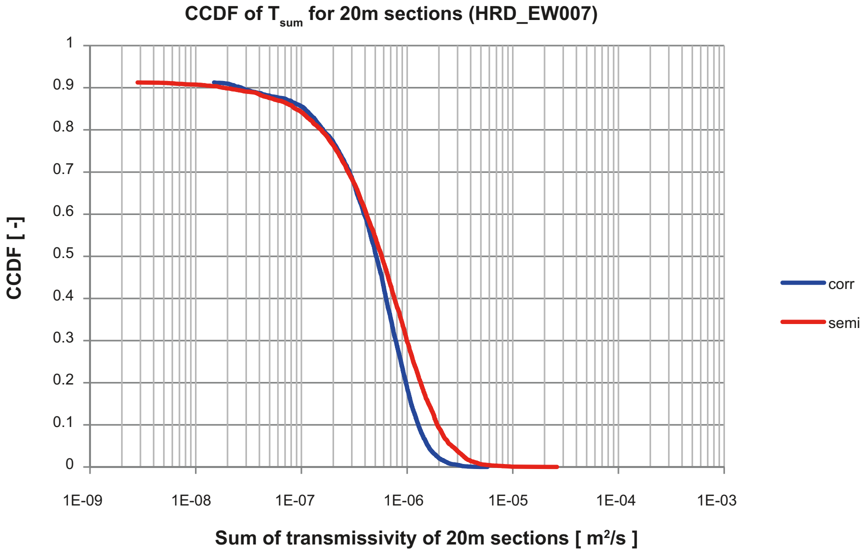

Figure 3-13. The complementary cumulative density functions of sum of transmissivity in horizontal 20 m sections for the correlated and semi correlated transmissivity model for HRD_ EW007 at the depth interval 400 to 650 m below sea level.

Table 3-13. The percentiles for the cumulative density function of sum of transmissivity in horizontal 20 m sections for HRD_ EW007 at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	2.70E–08	2.24E–07	5.09E–07	8.73E–07	1.28E–06	1.58E–06	6.14E–07	5.35E–07
Semi	–	1.81E–08	2.15E–07	5.76E–07	1.15E–06	1.93E–06	2.64E–06	8.58E–07	1.06E–06

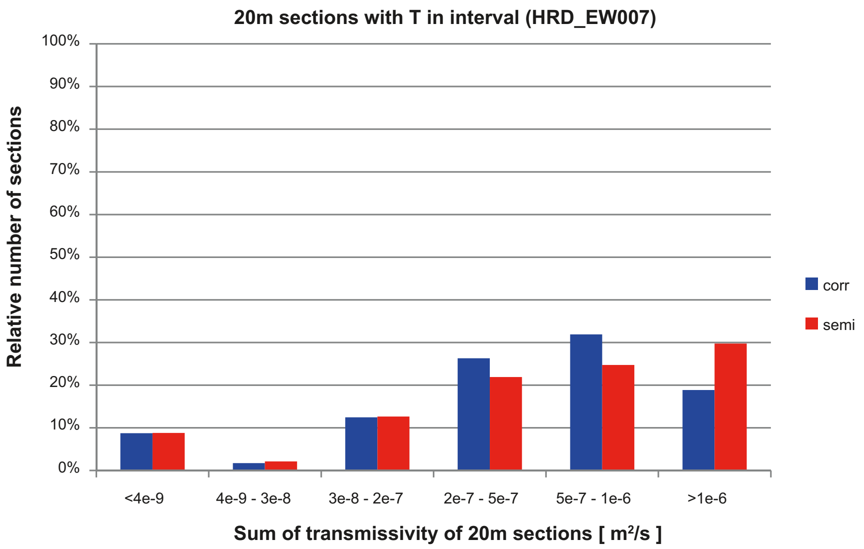

Figure 3-14. Histogram of the graph in Figure 3-13, i.e. the relative number of 20 m sections with total transmissivity in the given intervals.

Table 3-14. Tabulated numbers of Figure 3-14.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.0874	0.0172	0.1248	0.2630	0.3190	0.1886
semi	0.0880	0.0212	0.1266	0.2192	0.2474	0.2976

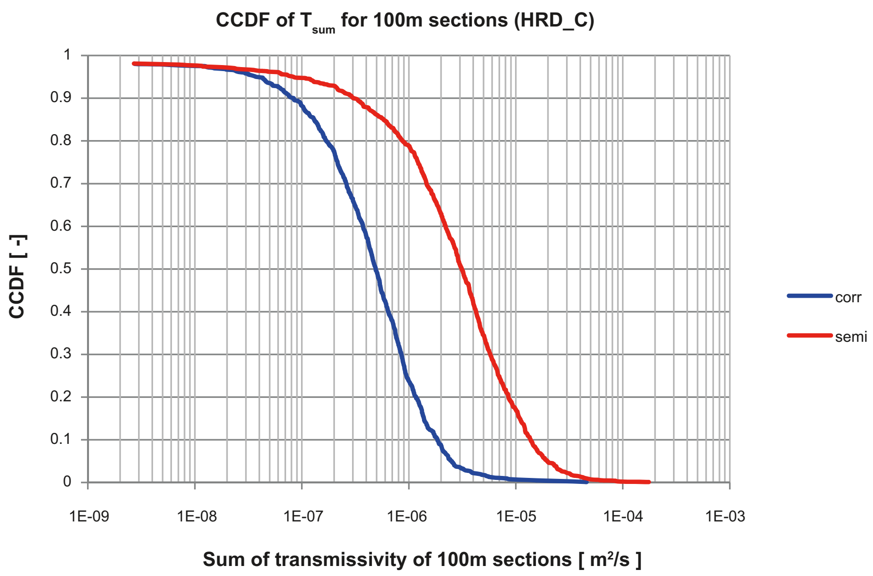

Figure 3-15. The complementary cumulative density functions of sum of transmissivity in horizontal 100 m sections for the correlated and semi correlated transmissivity model for HRD_C at the depth interval 400 to 650 m below sea level.

Table 3-15. The percentiles for the cumulative density function of sum of transmissivity in horizontal 100 m sections for HRD_C at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	3.73E–08	8.34E–08	2.16E–07	4.90E–07	9.47E–07	1.85E–06	2.46E–06	9.28E–07	2.39E–06
Semi	8.33E–08	3.01E–07	1.22E–06	3.11E–06	6.87E–06	1.37E–05	1.95E–05	5.95E–06	1.01E–05

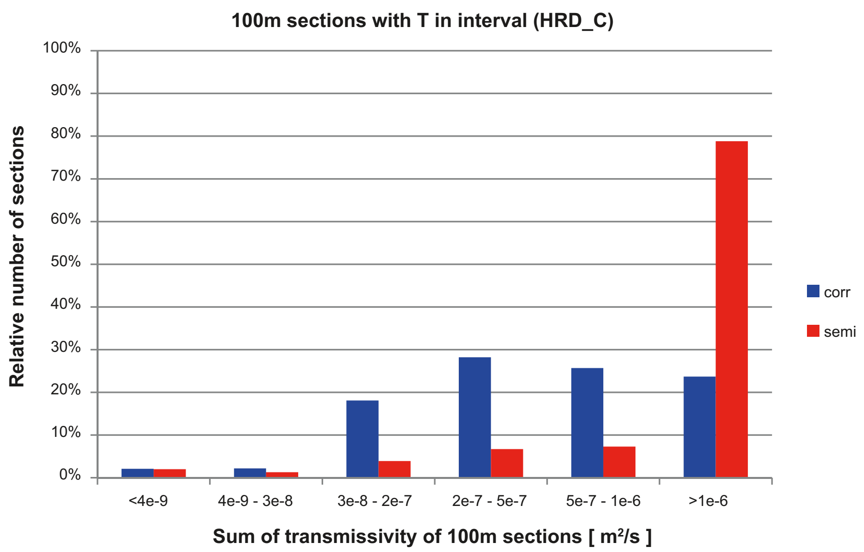

Figure 3-16. Histogram of the graph in Figure 3-15, i.e. the relative number of 100 m sections with total transmissivity in the given intervals.

Table 3-16. Tabulated numbers of Figure 3-16.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.021	0.022	0.181	0.282	0.257	0.237
semi	0.020	0.013	0.039	0.067	0.073	0.788

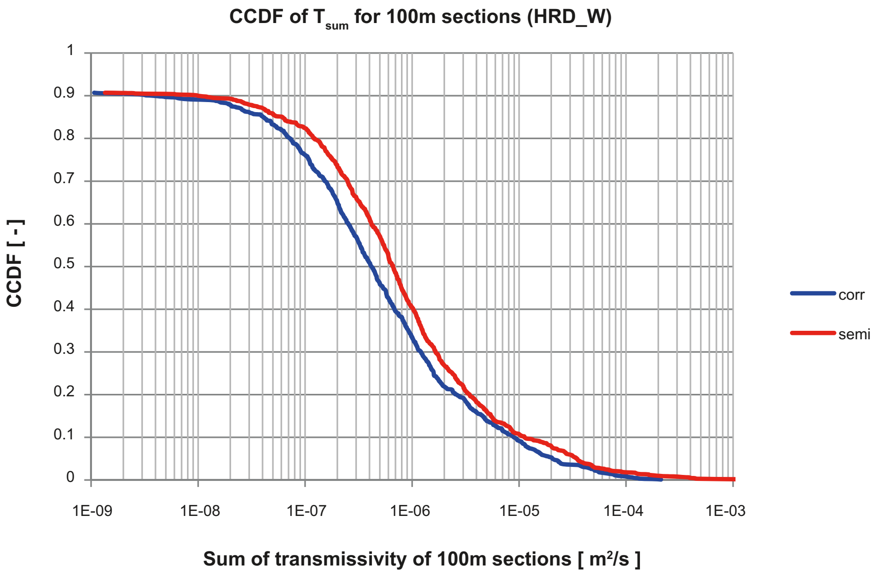

Figure 3-17. The complementary cumulative density functions of sum of transmissivity in horizontal 100 m sections for the correlated and semi correlated transmissivity model for HRD_W at the depth interval 400 to 650 m below sea level.

Table 3-17. The percentiles for the cumulative density function of sum of transmissivity in horizontal 100 m sections for HRD_W at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	3.77E–09	1.09E–07	4.15E–07	1.59E–06	8.86E–06	2.03E–05	4.49E–06	1.59E–05
Semi	–	9.93E–09	1.80E–07	6.63E–07	2.36E–06	1.12E–05	3.40E–05	1.10E–05	8.88E–05

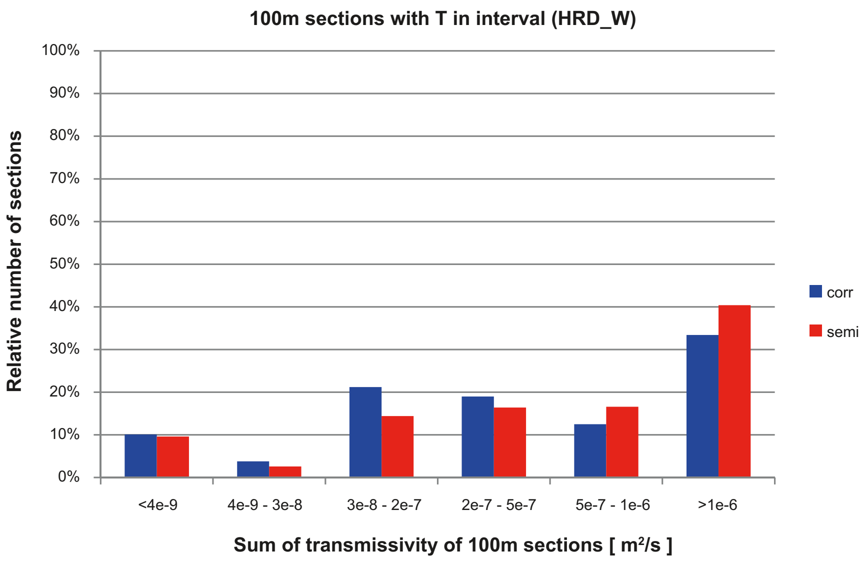

Figure 3-18. Histogram of the graph in Figure 3-17, i.e. the relative number of 100 m sections with total transmissivity in the given intervals.

Table 3-18. Tabulated numbers of Figure 3-18.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.101	0.038	0.212	0.19	0.125	0.334
semi	0.096	0.026	0.144	0.164	0.166	0.404

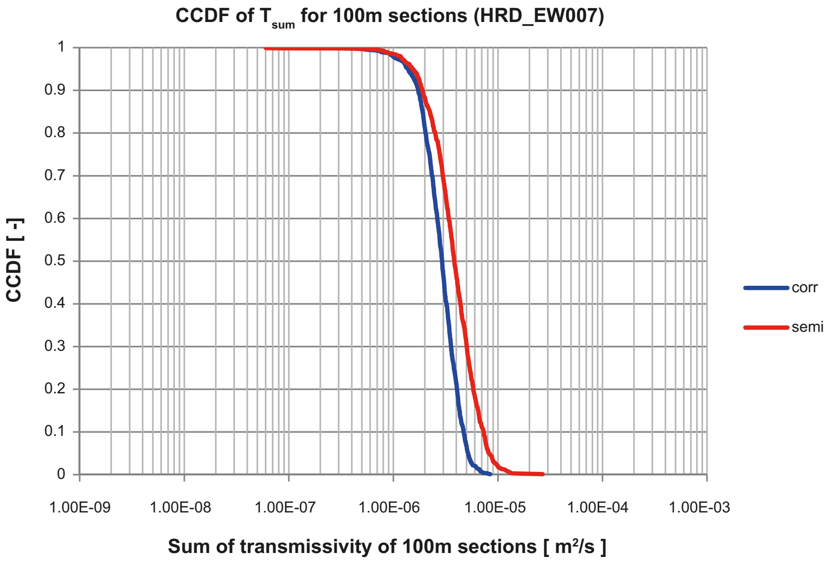

Figure 3-19. *The complementary cumulative density functions of sum of transmissivity in horizontal 100 m sections for the correlated and semi correlated transmissivity model for HRD_EW007 at the depth interval 400 to 650 m below sea level.*

Table 3-19. The percentiles for the cumulative density function of sum of transmissivity in horizontal 100 m sections for HRD_ EW007 at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	1.39E–06	1.73E–06	2.21E–06	2.93E–06	3.79E–06	4.70E–06	5.18E–06	3.07E–06	1.20E–06
Semi	1.54E–06	1.90E–06	2.79E–06	3.83E–06	5.37E–06	7.29E–06	8.18E–06	4.29E–06	2.29E–06

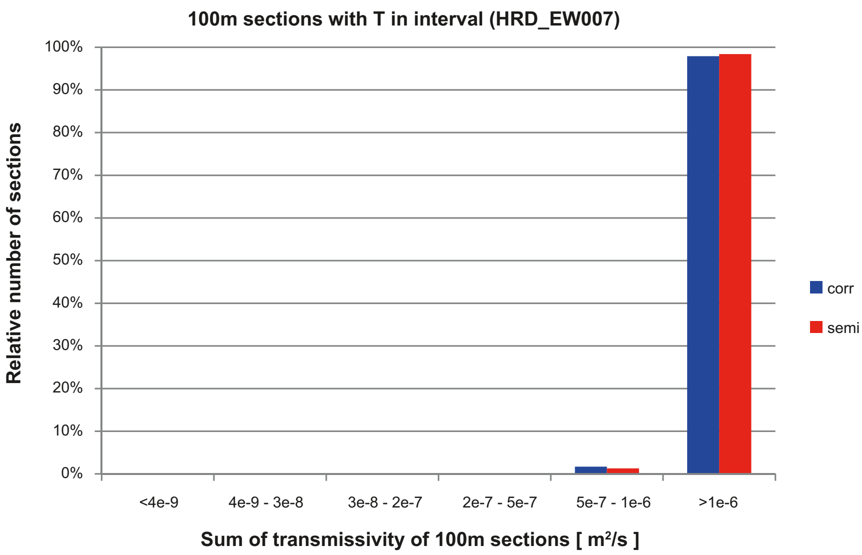

Figure 3-20. *Histogram of the graph in Figure 3-19, i.e. the relative number of 100 m sections with total transmissivity in the given intervals.*

Table 3-20. Tabulated numbers of Figure 3-20.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0	0	0.002	0.002	0.017	0.979
semi	0	0	0.002	0.001	0.013	0.984

3.2.2 Forsmark

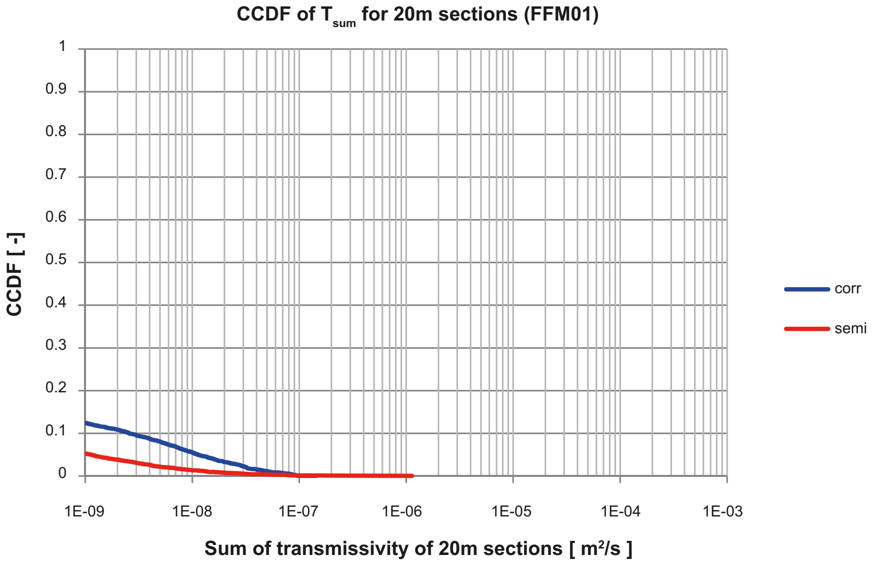

Figure 3-21. The complementary cumulative density functions of sum of transmissivity in horizontal 20 m sections for the correlated and semi correlated transmissivity model for FFM01 at the beneath below 400 m below sea level.

Table 3-21. The percentiles for the cumulative density function of sum of transmissivity in horizontal 20 m sections for FFM01 at the depth beneath 400 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	–	2.56E–09	1.12E–08	2.13E–09	9.53E–09
Semi	–	–	–	–	–	1.38E–10	1.13E–09	1.02E–09	1.94E–08

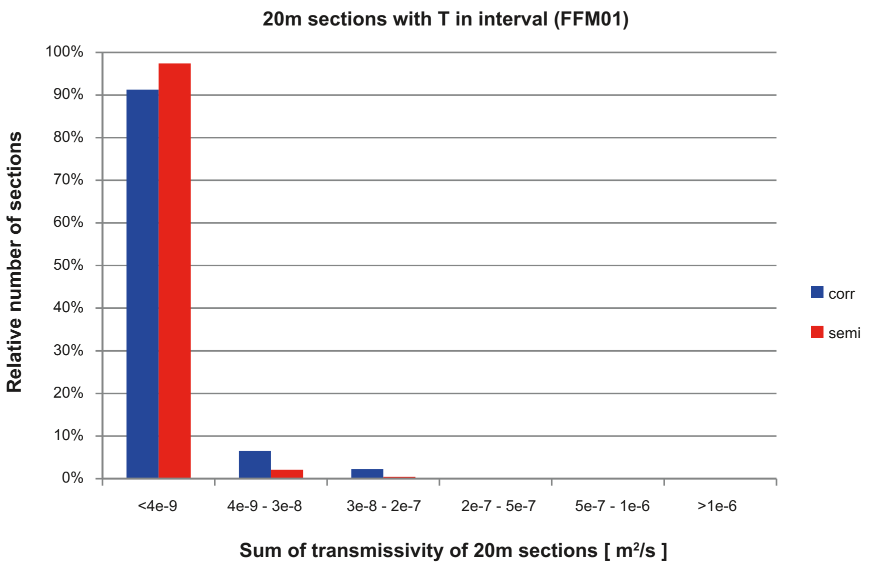

Figure 3-22. Histogram of the graph in Figure 3-21 , i.e. the relative number of 20 m sections with total transmissivity in the given intervals.

Table 3-22. Tabulated numbers of Figure 3-22.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.9128	0.0648	0.0224	0	0	0
semi	0.9742	0.0208	0.0042	0.0004	0.0002	0.0002

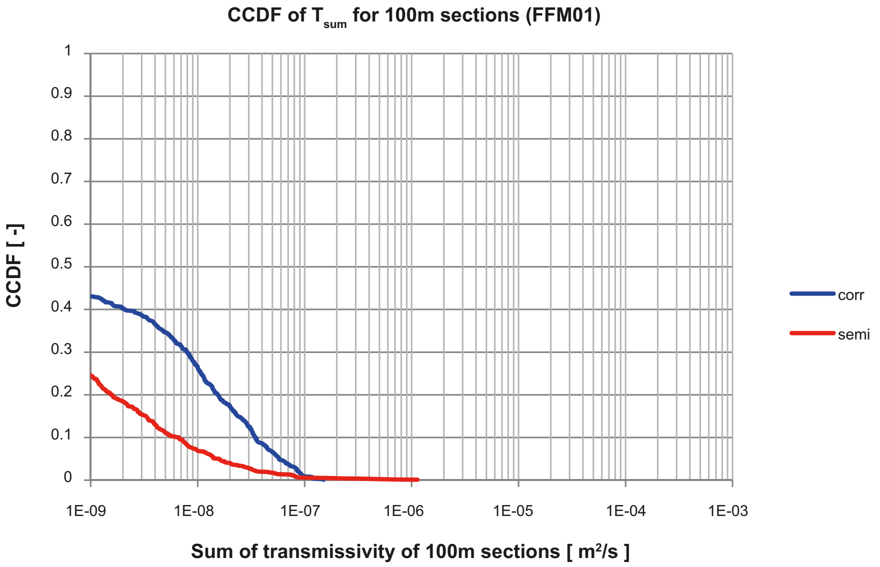

Figure 3-23. *The complementary cumulative density functions of sum of transmissivity in horizontal 100 m sections for the correlated and semi correlated transmissivity model for FFM01 at the beneath below 400 m below sea level.*

Table 3-23. The percentiles for the cumulative density function of sum of transmissivity in horizontal 100 m sections for FFM01 at the depth beneath 400 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	1.07E–08	3.46E–08	5.74E–08	1.07E–08	2.15E–08
Semi	–	–	–	–	9.03E–10	6.25E–09	1.42E–08	5.09E–09	4.33E–08

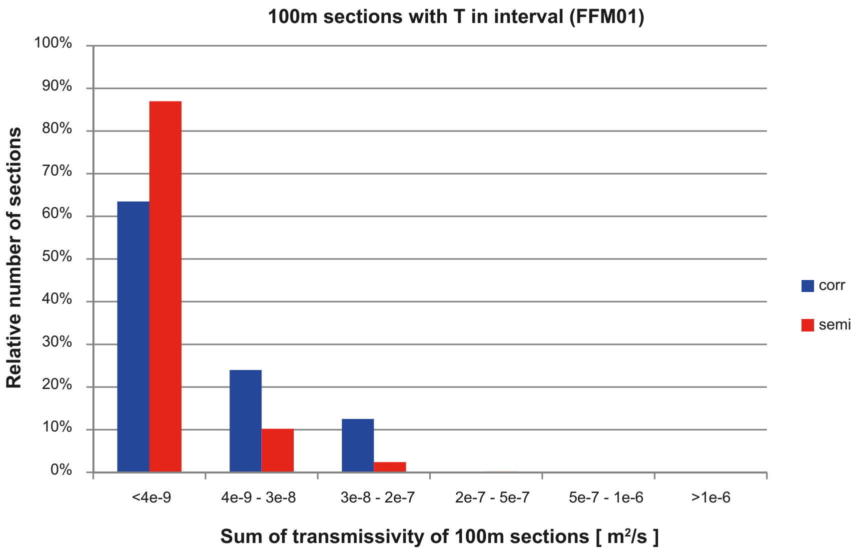

Figure 3-24. *Histogram of the graph in Figure 3-23 , i.e. the relative number of 100 m sections with total transmissivity in the given intervals.*

Table 3-24. Tabulated numbers of Figure 3-24.

model	$<4{\cdot}10^{-9}$	$4{\cdot}10^{-9}$–$3{\cdot}10^{-8}$	$3{\cdot}10^{-8}$–$2{\cdot}10^{-7}$	$2{\cdot}10^{-7}$–$5{\cdot}10^{-7}$	$5{\cdot}10^{-7}$–$1{\cdot}10^{-6}$	$>1{\cdot}10^{-6}$
corr	0.635	0.24	0.125	0	0	0
semi	0.87	0.102	0.024	0.002	0.001	0.001

3.3 Total transmissivity in deposition holes

This chapter shows statistics of the sum of transmissivity of all features intersecting 8 m vertical sections. There are 12,500 sections that the statistics is calculated from. The data is shown as Complementary Cumulative Density Functions, CCDF, i.e. the probability that the sum of the transmissivity of the fractures in a section is larger than a specific value. The tables below each graph shows different percentiles for the Cumulative Density Function, CDF, i.e. the probability that the sum of transmissivities of intersecting fractures in a section is smaller than the, in the table, given value. Some sections are not intersected by any fracture, or the sum of transmissivities is below $1 \cdot 10^{-9}$ m^2/s and hence the graphs do not always start at 1.

To facilitate the reading of the CCDF graphs a histogram is provided where the relative number of sections in a specific transmissivity interval is shown. Below each histogram the values of the bars are presented in a table.

3.3.1 Laxemar

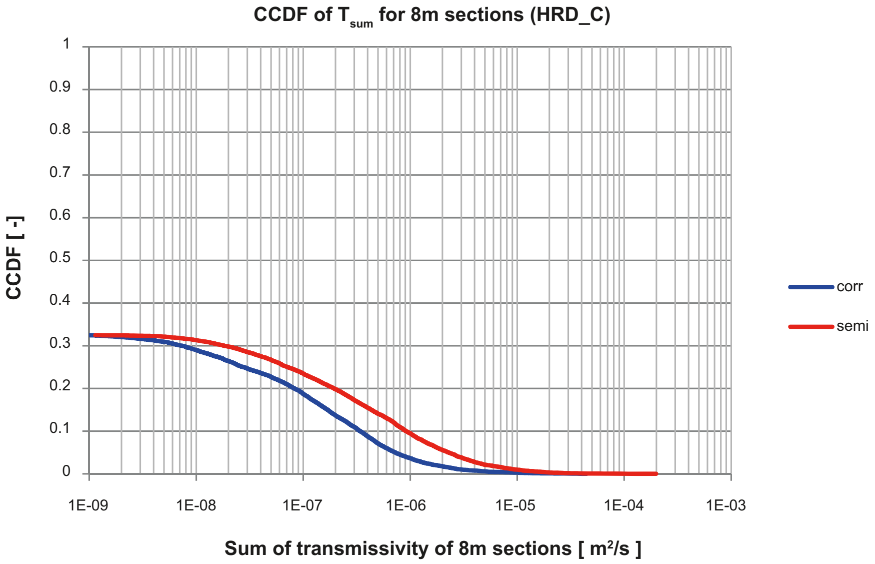

Figure 3-25. *The complementary cumulative density functions of sum of transmissivity in vertical 8 m sections for the correlated and semi correlated transmissivity model for HRD_C at the depth interval 400 to 650 m below sea level.*

Table 3-25. The percentiles for the cumulative density function of sum of transmissivity in vertical 8 m sections for HRD_C at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	2.76E–08	3.40E–07	7.25E–07	1.92E–07	1.26E–06
Semi	–	–	–	–	7.15E–08	9.14E–07	2.27E–06	5.31E–07	3.44E–06

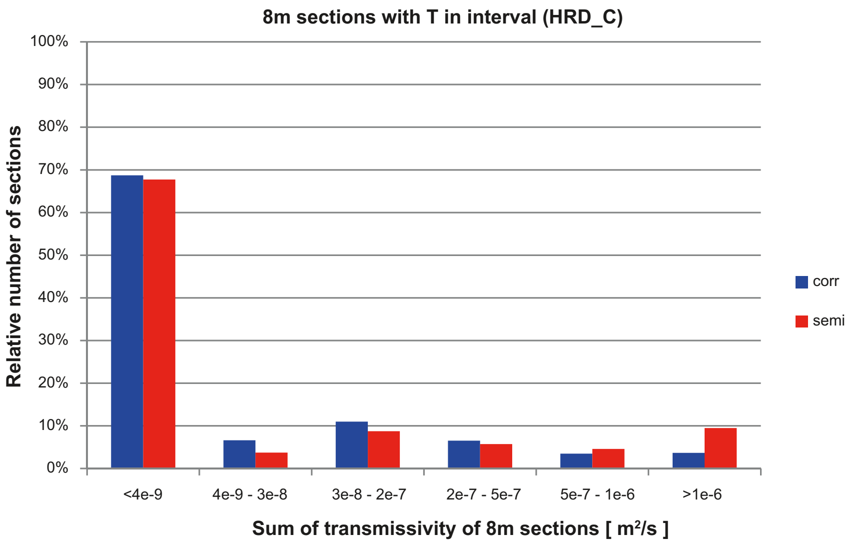

Figure 3-26. *Histogram of the graph in Figure 3-25, i.e. the relative number of 8 m sections with total transmissivity in the given intervals.*

Table 3-26. Tabulated numbers of Figure 3-26.

model	$<4 \cdot 10^{-9}$	$4 \cdot 10^{-9}$–$3 \cdot 10^{-8}$	$3 \cdot 10^{-8}$–$2 \cdot 10^{-7}$	$2 \cdot 10^{-7}$–$5 \cdot 10^{-7}$	$5 \cdot 10^{-7}$–$1 \cdot 10^{-6}$	$>1 \cdot 10^{-6}$
corr	0.68736	0.06616	0.10984	0.06536	0.03472	0.03656
semi	0.67728	0.03728	0.08744	0.0572	0.04592	0.09488

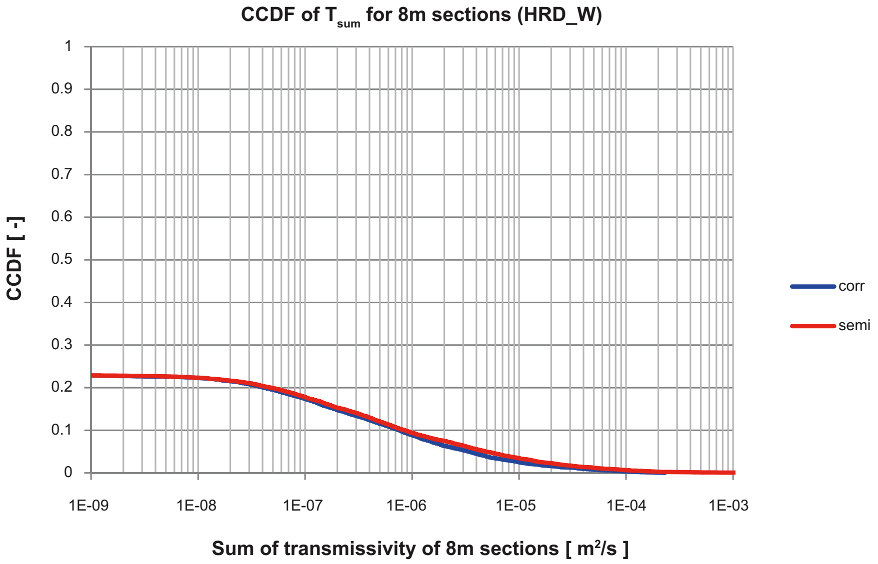

Figure 3-27. The complementary cumulative density functions of sum of transmissivity in vertical 8 m sections for the correlated and semi correlated transmissivity model for HRD_W at the depth interval 400 to 650 m below sea level.

Table 3-27. The percentiles for the cumulative density function of sum of transmissivity in vertical 8 m sections for HRD_W at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	–	7.59E–07	3.35E–06	1.45E–06	1.06E–05
Semi	–	–	–	–	–	8.53E–07	4.85E–06	3.78E–06	6.12E–05

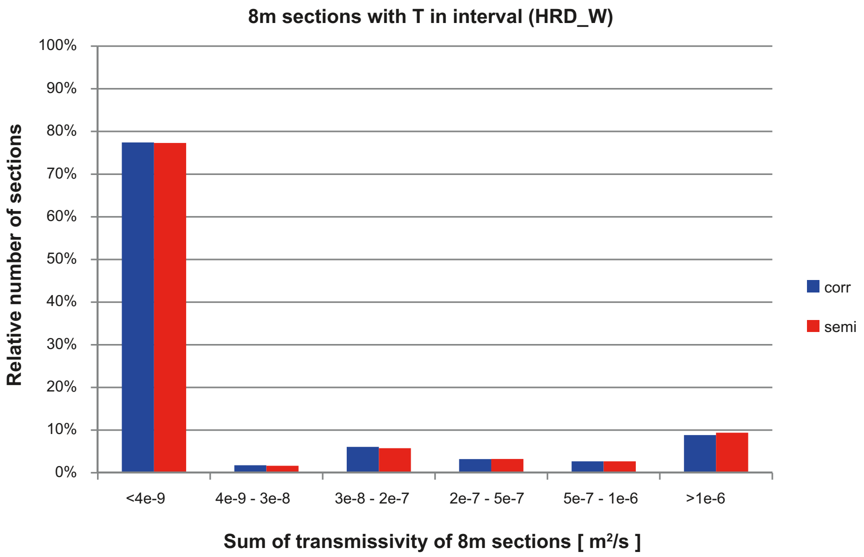

Figure 3-28. Histogram of the graph in Figure 3-27, i.e. the relative number of 8 m sections with total transmissivity in the given intervals.

Table 3-28. Tabulated numbers of Figure 3-28.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.77424	0.01768	0.06072	0.03192	0.02696	0.08848
semi	0.77296	0.01648	0.05768	0.03224	0.02672	0.09392

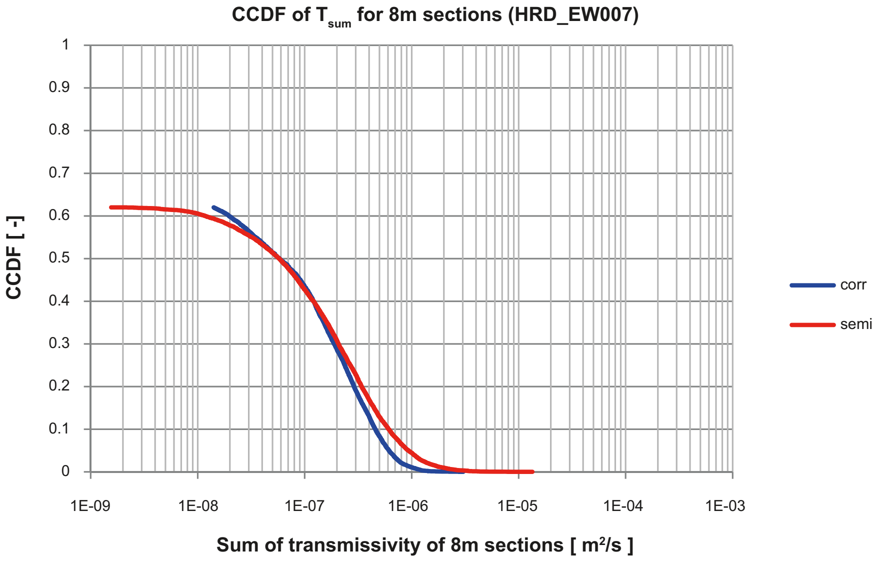

Figure 3-29. *The complementary cumulative density functions of sum of transmissivity in vertical 8 m sections for the correlated and semi correlated transmissivity model for HRD_EW007 at the depth interval 400 to 650 m below sea level.*

Table 3-29. The percentiles for the cumulative density function of sum of transmissivity in vertical 8 m sections for HRD_EW007 at the depth interval 400 to 650 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	5.81E–08	2.37E–07	4.57E–07	6.13E–07	1.58E–07	2.30E–07
Semi	–	–	–	5.78E–08	2.67E–07	6.10E–07	9.30E–07	2.18E–07	4.35E–07

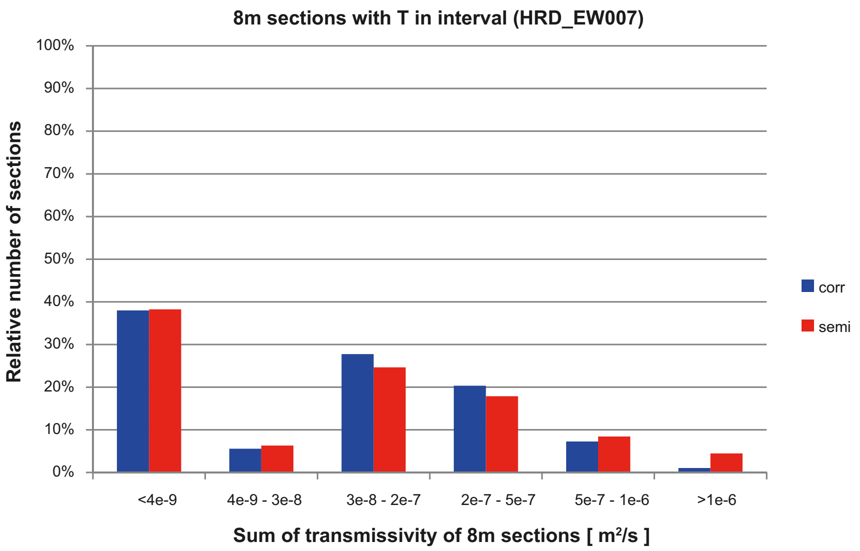

Figure 3-30. *Histogram of the graph in Figure 3-29, i.e. the relative number of 8 m sections with total transmissivity in the given intervals.*

Table 3-30. Tabulated numbers of Figure 3-30.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.38008	0.05576	0.27744	0.20344	0.07264	0.01064
semi	0.38232	0.0632	0.24656	0.17888	0.08448	0.04456

3.3.2 Forsmark

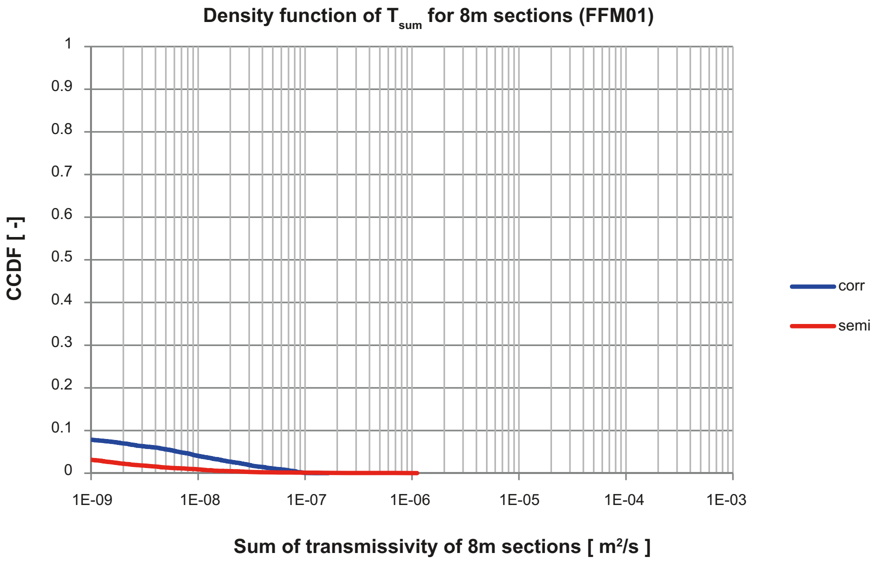

Figure 3-31. The complementary cumulative density functions of sum of transmissivity in vertical 8 m sections for the correlated and semi correlated transmissivity model for FFM01beneath the depth 400 m below sea level.

Table 3-31. The percentiles for the cumulative density function of sum of transmissivity in vertical 8 m sections for FFM01beneath the depth 400 m below sea level.

Model	5%	10%	25%	50%	75%	90%	95%	average	std dev
Corr	–	–	–	–	–	–	6.56E–09	1.70E–09	9.16E–09
Semi	–	–	–	–	–	–	2.52E–10	7.10E–10	1.82E–08

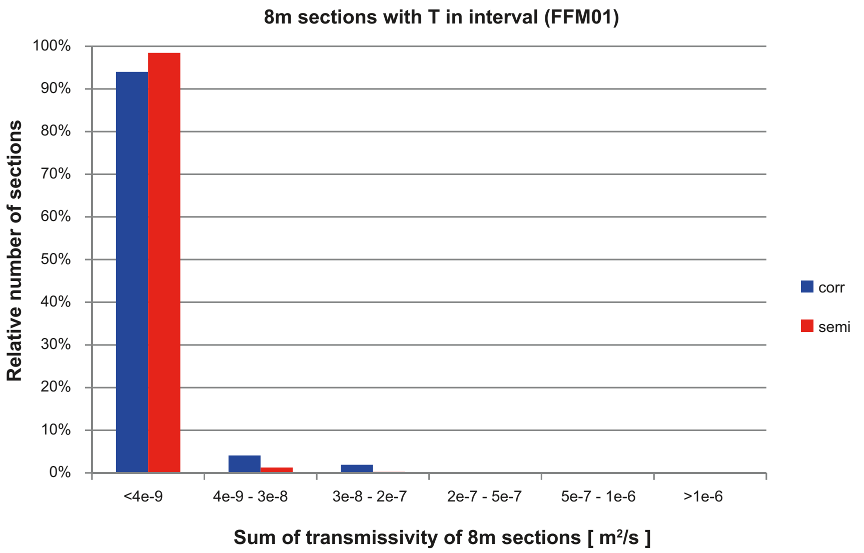

Figure 3-32. Histogram of the graph in Figure 3-31, i.e. the relative number of 8 m sections with total transmissivity in the given intervals.

Table 3-32. Tabulated numbers of Figure 3-32.

model	$<4\cdot10^{-9}$	$4\cdot10^{-9}$–$3\cdot10^{-8}$	$3\cdot10^{-8}$–$2\cdot10^{-7}$	$2\cdot10^{-7}$–$5\cdot10^{-7}$	$5\cdot10^{-7}$–$1\cdot10^{-6}$	$>1\cdot10^{-6}$
corr	0.94008	0.04088	0.01904	0	0	0
semi	0.98456	0.01264	0.00232	0.00024	0	0.00024

References

Follin S, Johansson P, Hartley L, Jackson P, Roberts D, Marsic N, 2007a. Hydrogeological conceptual model development and numerical modelling using CONNECTFLOW. Forsmark modelling stage 2.2, SKB R-07-49, Svensk Kärnbränslehantering AB.

Follin S, Levén J, Hartley L, Jackson P, Joyce S, Roberts D, Swift B, 2007b. Hydrogeological characterisation and modelling of deformation zones and fracture domains, Forsmark modelling stage 2.2. SKB R-07-48, Svensk Kärnbränslehantering AB.

Hansson B, Magnusson J, Söderlund P, 2009a. Underground design Forsmark, Layout D2, Layout and construction plan. SKB R-08-113, Svensk Kärnbränslehantering AB.

Hansson B, Magnusson J, Söderlund P, 2009b. Underground Design Laxemar, Design D2 Layout and Construction Plan. SKB R-09-08, Svensk Kärnbränslehantering AB.

Rhén I, Forsmark T, Hartley L, Jackson P, Roberts D, Swan D, Gylling B, 2008. Hydrogeological conceptualisation and parameterisation, Site descriptive modelling SDM-Site Laxemar, SKB R-08-78, Svensk Kärnbränslehantering AB. In prep.

SKB, 2007. Final repository facility, Underground design premises/D2. SKB R-07-33, Svensk Kärnbränslehantering AB.

SKB, 2008a. Forsmark, Site Engineering Report, Guidelines for Underground Design, Step D2. SKB R-08-83, Svensk Kärnbränslehantering AB. In prep.

SKB, 2008b. Laxemar, Site Engineering Report, Guidelines for Underground Design, Step D2. SKB R-08-88, Svensk Kärnbränslehantering AB. In prep.

Svensson U, Kuylenstierna H-O, Ferry M, 2008. DarcyTools, Version 3.2. Documentation to be printed during 2009.

Appendix A

Source codes developed for the analyses

A.1 Sampling of Fracture Network by scanline (Intel Fortran 11)

```
!====================================================================!
!====================================================================!
!===============           DT_BH_sampling.f           ===============!
!====================================================================!
!====================================================================!
!====                                                            ====!
!===  Console application working with DarcyTools V3.2 files.     ===!
!===                                                               ===!
!===  Program that finds out the number of intersections between  ===!
!===  lines and fractures. A maximum of 10 lines with maximum 350  ===!
!===  vertices each are allowed. The number of fracture files are  ===!
!===  unlimited, however each of the 10 lines can not intersect    ===!
!===  more than 10 000 fractures                                   ===!
!====                                                            ====!
!====================================================================!
!====                                                            ====!
!===  INPUT:                                                       ===!
!===    Name of the file containing the path(s) to the borehole    ===!
!===    file(s)                                                    ===!
!===    Name of the file containing the path(s) to the fracture    ===!
!===    file(s)                                                    ===!
!===                                                               ===!
!===    The file containing the path(s) to the borehole file(s)    ===!
!===    starts with a figure telling the number of files to be     ===!
!===    read then the names of the files come, one at a row, e.g.  ===!
!===                                                               ===!
!===      2                                                        ===!
!===      myFirstBoreHole.dat                                      ===!
!===      mySecondBoreHole.dat                                     ===!
!===                                                               ===!
!===    A maximum of 10 boreholes are allowed                      ===!
!===                                                               ===!
!===    The Boreholefiles starts with a number telling the number  ===!
!===    of vertices for the hole and then comes each vertex on a   ===!
!===    single row, i.e. the format is like                        ===!
!===                                                               ===!
!===      4                                                        ===!
!===      -154.000, 120.000, 0.000                                 ===!
!===      -543.873, 120.134, -2.987                                ===!
!===      -543.734, 120.214, -5.870                                ===!
!===      -543.598, 120.296, -8.731                                ===!
!====                                                            ====!
!====================================================================!
!====                                                            ====!
!===  OUTPUT:                                                      ===!
!===                                                               ===!
!====================================================================!
!===                                                               ===!
!===  Written by Martin Stigsson SKB Jan 29 2009                    ===!
!===                                                               ===!
!====================================================================!

      program DTreadFracFile

c     Make sure that all variables not declared are treated as logical
      IMPLICIT LOGICAL (a-z)

c     Definition of constants, in order:
c                max number of scanlines, i.e. # of files with a line
c                Max number of vertices in each line
c                Max number of intersecting fractures to each line
c                Maximum length of file names
      parameter (maxNofSL=10,
     &           maxNofVert=350,
     &           maxNofFrax=10000,
     &           maxFileNameLength=50)

c     Declarations
      character fileheader*8
      character*(maxFileNameLength) slPaths, fracFilePaths, slFile,
     &                             fracFile,outputFile
```

```
      logical inhomo
      integer errorMarker, i, ii, j, jj, k, nofSL, nofFracFiles
      integer ns, nv, setNo

c     Containing the number of vertices along each Scan line
      integer nofVert(1:maxNofVert)
c     Contains all the vertices of each scan line
      real lineVert(1:maxNofSL, 1:maxNofVert, 1:3)
c     Contains the number of fractures that is crossing each scanline
      integer nofHits(1:maxNofSL)
c     Contains all the data about each saved fracture, in the order of
c     appearence in the binary fracture file
           real fracMatrix(1:maxNofSL, 1:maxNofFrax, 1:59)
c     Contains the set name for each saved fracture
           integer fracSetMatrix(1:maxNofSL, 1:maxNofFrax)
c     Rotation matrix
      real aMatr(1:3, 1:3)
c     The equation of the plane values for 1 fracture
      real A, B, C, D

c     Vectors containing the values for the equation of the plane of the
c     fracture box
      real Av(6), Bv(6), Cv(6), Dv(6)

c     Vectors containing the corners of the fracture box
      real Xv(8), Yv(8), Zv(8)
c     Data stored for each fracture
           real qxkl, qxkh, qykl, qykh, qzkl, qzkh, vo, su, st, co, di
c     Corner co-ordinates of the fracture midplane
           real x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4
c     Coordinates of the intersection point between line and plane
           real xc,yc,zc
c     Length of the side of the fracture
           real v1L, v2L
c     intersection point together with max and min points of fracture
c     extention in rotated co-ordinate system
      real xPc, yPc, xPmin, yPmin, zPmin, xPmax, yPmax, zPmax
c     Numbers for calculating the equation of the line
      real t, num, denom

c     Read the name of the file containing the paths to the scan lines
      write(6,*) 'Enter the name of the file containing
     &  the paths to scanline files'
      read(5,*) slPaths
      write(6,*) slPaths

c     Read the name of the file containing the paths to the fracture
c     files
      write(6,*) 'Enter the name of the file containing
     &  the paths to fracture files'
      read(5,*) fracFilePaths
      write(6,*) fracFilePaths



c     Open the file containing the paths to the scanline files
      open (70, file=slPaths, form='formatted', IOSTAT=errorMarker)
      if (errorMarker.ne.0) then
        write (6, *) 'Could not open input file, aborting.'
        goto 990
      endif

c     Read the number of scan line files that will be investigated
      read(70,*) nofSL
      write(6,*) 'Number of files to read: ', nofSL

c     Open, read and save the data for each scan line file
      do i=1,nofSL

c       Read the name of the scan line file
        read(70,*) slFile
        write(6,*) 'open file: ', slFile

c       Open the scan line file
        open (71, file=slFile, form='formatted', IOSTAT=errorMarker)
        if (errorMarker.ne.0) then
          write (6, *) 'Could not open input file, aborting.'
          goto 990
        endif
```

```fortran
c       Read the number of vertices to come
        read(71,*) nofVert(i)
        write(6,*) 'number of vertices in scanline: ', nofVert(i)

        write(6,*) '      X              Y              Z'
c       Save
        do  j=1,nofVert(i)
          read(71, *) lineVert(i,j,1),lineVert(i,j,2),lineVert(i,j,3)
          write(6, *) lineVert(i,j,1),lineVert(i,j,2),lineVert(i,j,3)
        end do

        close (71, iostat=errorMarker)

      end do

      close (70, iostat=errorMarker)



c     Open the file containing the paths to the fracture files
      open (72, file=fracFilePaths, form='formatted',IOSTAT=errorMarker)
      if (errorMarker.ne.0) then
        write (6, *) 'Could not open input file, aborting.'
        goto 990
      endif

c     Read the number of Fracture files that will be investigated
      read(72,*) nofFracFiles
      write(6,*) 'Number of files to read: ', nofFracFiles

c     Open, read and save the data for each fracture that is crossing
c     the scanline
      do i=1,nofFracFiles

c       Read the name of the fracture file
        read(72,*) fracFile
        write(6,*) 'open file: ', fracFile


c       Open the fracture file
        open (73, file=fracFile, form='unformatted', IOSTAT=errorMarker)
        if (errorMarker.ne.0) then
          write (6, *) 'Could not open input file, aborting.'
          goto 990
        endif


c       Read and check that the header file is correct
        read (73, iostat=errorMarker, end=990) fileheader

        if (errorMarker.ne.0) then
           write (6, *) 'Error reading from input file, aborting.'
           goto 990
        else if (fileheader.ne.'#FR#V310') then
           write (6, *) 'Wrong header in input file, aborting.'
           goto 990
        else
           write (6, *) 'Right header in input file, continue.'
        end if



c       Read the fracture file to the end
        do

c         Read One fracture
          read (73, iostat=errorMarker, end=40) ns, nv, inhomo, setNo,
     &    (Av(j), Bv(j), Cv(j), Dv(j), j=1, ns),
     &    (Xv(k), Yv(k), Zv(k), k=1, nv),
     &    qxkl, qxkh, qykl, qykh, qzkl, qzkh, vo, su, st, co, di


c         Calculate the midplane of the fracture
          x1=0.5*(Xv(1)+Xv(2))
          y1=0.5*(Yv(1)+Yv(2))
          z1=0.5*(Zv(1)+Zv(2))
```

```
          x2=0.5*(Xv(3)+Xv(4))
          y2=0.5*(Yv(3)+Yv(4))
          z2=0.5*(Zv(3)+Zv(4))
          x3=0.5*(Xv(5)+Xv(6))
          y3=0.5*(Yv(5)+Yv(6))
          z3=0.5*(Zv(5)+Zv(6))
          x4=0.5*(Xv(7)+Xv(8))
          y4=0.5*(Yv(7)+Yv(8))
          z4=0.5*(Zv(7)+Zv(8))

c         Save and calculate the equation of the mid plane
          A=Av(1)
          B=Bv(1)
          C=Cv(1)
          D=-(A*x1 + B*y1 + C*z1)

c         Go through each borehole
          do j=1,nofSL

c           Go through each segment of the scanline
            do k=1,nofVert(j)-1

c             The equation of the line is
c              + X=P1x+(P2x-P1x)*t
c              | Y=P1y+(P2y-P1y)*t
c              + Z=P1z+(P2z-P1z)*t
c
c             The equation of the plane is
c              Ax+Bx+cx+D=0
c
c             Substituting the x, y and z in the equation of the plane
c             with the x, y and z in the equation of the line and
c             solving for t gives
c
c                  -(D + A*P1x + B*P1y + C*P1z)
c              t=--------------------------------
c                A(P2x-P1x)+B(P2y-P1y)+C(P2z-P1z)

c             Calculate the numerator
              num= -(D+
     +               A*lineVert(j,k,1)+
     +               B*lineVert(j,k,2)+
     +               C*lineVert(j,k,3))

c             Calculate the denominator
              denom=A*(lineVert(j,k+1,1)-lineVert(j,k,1))+
     +              B*(lineVert(j,k+1,2)-lineVert(j,k,2))+
     +              C*(lineVert(j,k+1,3)-lineVert(j,k,3))

c             If the denominator <> 0 the line segment is non-paralell
c             to the fracture plane, and an intersection occur between
c             the segment and the plane if 0<t<1. If the nominator =0
c             the line segement and plane are parallel, and hence no
c             intersection occur unless the segemnt resides in the
c             plane. This case is neglected and t is set a dummy value
c             of 100 to discriminate it from further calculations.

              if (denom .ne. 0) then
                t=num/denom
              else
                t=100
              endif

c             if 0<t<1 the intersection is between the 2 point of the
c             line segment and potential intersection exist
              if (0 .le. t .and. t .lt. 1) then

c               Calculate the crossing point
                xc=lineVert(j,k,1)+(lineVert(j,k+1,1)-lineVert(j,k,1))*t
                yc=lineVert(j,k,2)+(lineVert(j,k+1,2)-lineVert(j,k,2))*t
                zc=lineVert(j,k,3)+(lineVert(j,k+1,3)-lineVert(j,k,3))*t

c
c               Calculate the matrix for transforming the
c               coordinate system to the fracture plane and
c               the coefficients for the equation of the plane
c
c                         ------>V1
c                    1______________2
```

```
c                    /              /
c                /  /              /
c               /  /              /
c            ¦ /  /              /
c            +-- /              /
c           V2  /______________/
c              3               4
c
c               Calculate the length of the sides of the plane
                v1L=sqrt((x2-x1)**2+(y2-y1)**2+(z2-z1)**2)
                v2L=sqrt((x3-x1)**2+(y3-y1)**2+(z3-z1)**2)

c
c               Make the vector between point 1 and 2 the x' axis
                aMatr(1,1)=(x2-x1)/v1L
                aMatr(1,2)=(y2-y1)/v1L
                aMatr(1,3)=(z2-z1)/v1L
c
c               Make the vector between point 1 and 3 the y' axis
                aMatr(2,1)=(x3-x1)/v2L
                aMatr(2,2)=(y3-y1)/v2L
                aMatr(2,3)=(z3-z1)/v2L
c
c               Get the z' axis Using the Equation of the plane ionstead
c               of Cross product
                aMatr(3,1)=A
                aMatr(3,2)=B
                aMatr(3,3)=C

                xPmin=aMatr(1,1)*x1+aMatr(1,2)*y1+aMatr(1,3)*z1
                yPmin=aMatr(2,1)*x1+aMatr(2,2)*y1+aMatr(2,3)*z1

                xPmax=aMatr(1,1)*x4+aMatr(1,2)*y4+aMatr(1,3)*z4
                yPmax=aMatr(2,1)*x4+aMatr(2,2)*y4+aMatr(2,3)*z4

                xPc=aMatr(1,1)*xc+aMatr(1,2)*yc+aMatr(1,3)*zc
                yPc=aMatr(2,1)*xc+aMatr(2,2)*yc+aMatr(2,3)*zc

                if(xPmin .le. xPc) then
                 if (xPmax .ge. xPc) then
                  if (yPmin .le. yPc) then
                   if (yPmax .ge. yPc) then


c                   Save the data of the fracture

                    nofHits(j) = nofHits(j)+1
                    fracSetMatrix(j, nofHits(j))=setNo
                    fracMatrix(j, nofHits(j), 1)=Av(1)
                    fracMatrix(j, nofHits(j), 2)=Bv(1)
                    fracMatrix(j, nofHits(j), 3)=Cv(1)
                    fracMatrix(j, nofHits(j), 4)=Dv(1)
                    fracMatrix(j, nofHits(j), 5)=Av(2)
                    fracMatrix(j, nofHits(j), 6)=Bv(2)
                    fracMatrix(j, nofHits(j), 7)=Cv(2)
                    fracMatrix(j, nofHits(j), 8)=Dv(2)
                    fracMatrix(j, nofHits(j), 9)=Av(3)
                    fracMatrix(j, nofHits(j), 10)=Bv(3)
                    fracMatrix(j, nofHits(j), 11)=Cv(3)
                    fracMatrix(j, nofHits(j), 12)=Dv(3)
                    fracMatrix(j, nofHits(j), 13)=Av(4)
                    fracMatrix(j, nofHits(j), 14)=Bv(4)
                    fracMatrix(j, nofHits(j), 15)=Cv(4)
                    fracMatrix(j, nofHits(j), 16)=Dv(4)
                    fracMatrix(j, nofHits(j), 17)=Av(5)
                    fracMatrix(j, nofHits(j), 18)=Bv(5)
                    fracMatrix(j, nofHits(j), 19)=Cv(5)
                    fracMatrix(j, nofHits(j), 20)=Dv(5)
                    fracMatrix(j, nofHits(j), 21)=Av(6)
                    fracMatrix(j, nofHits(j), 22)=Bv(6)
                    fracMatrix(j, nofHits(j), 23)=Cv(6)
                    fracMatrix(j, nofHits(j), 24)=Dv(6)

                    fracMatrix(j, nofHits(j), 25)=Xv(1)
                    fracMatrix(j, nofHits(j), 26)=Yv(1)
                    fracMatrix(j, nofHits(j), 27)=Zv(1)
                    fracMatrix(j, nofHits(j), 28)=Xv(2)
                    fracMatrix(j, nofHits(j), 29)=Yv(2)
                    fracMatrix(j, nofHits(j), 30)=Zv(2)
```

```
                  fracMatrix(j, nofHits(j), 31)=Xv(3)
                  fracMatrix(j, nofHits(j), 32)=Yv(3)
                  fracMatrix(j, nofHits(j), 33)=Zv(3)
                  fracMatrix(j, nofHits(j), 34)=Xv(4)
                  fracMatrix(j, nofHits(j), 35)=Yv(4)
                  fracMatrix(j, nofHits(j), 36)=Zv(4)
                  fracMatrix(j, nofHits(j), 37)=Xv(5)
                  fracMatrix(j, nofHits(j), 38)=Yv(5)
                  fracMatrix(j, nofHits(j), 39)=Zv(5)
                  fracMatrix(j, nofHits(j), 40)=Xv(6)
                  fracMatrix(j, nofHits(j), 41)=Yv(6)
                  fracMatrix(j, nofHits(j), 42)=Zv(6)
                  fracMatrix(j, nofHits(j), 43)=Xv(7)
                  fracMatrix(j, nofHits(j), 44)=Yv(7)
                  fracMatrix(j, nofHits(j), 45)=Zv(7)
                  fracMatrix(j, nofHits(j), 46)=Xv(8)
                  fracMatrix(j, nofHits(j), 47)=Yv(8)
                  fracMatrix(j, nofHits(j), 48)=Zv(8)

                  fracMatrix(j, nofHits(j), 49)=qxkl
                  fracMatrix(j, nofHits(j), 50)=qxkh
                  fracMatrix(j, nofHits(j), 51)=qykl
                  fracMatrix(j, nofHits(j), 52)=qykh
                  fracMatrix(j, nofHits(j), 53)=qzkl
                  fracMatrix(j, nofHits(j), 54)=qzkh
                  fracMatrix(j, nofHits(j), 55)=vo
                  fracMatrix(j, nofHits(j), 56)=su
                  fracMatrix(j, nofHits(j), 57)=st
                  fracMatrix(j, nofHits(j), 58)=co
                  fracMatrix(j, nofHits(j), 59)=di
                 endif
                endif
               endif
              endif

c             End of potential intersection,
c             i.e. if (0 .le. t .and. t .lt. 1) then
              endif

c           End of Go through each segment of the scan line,
c           i.e. do k=1,nofVert(j)-1
            end do

c         End of Go through each borehole, i.e. do j=1,nofSL
          end do

c       End of Read the fracture file to the end
        end do
c       When the end of the file is reached jump to here
40      continue


c     end of Open, read and save the data for each fracture that is
c     crossing the scanline, i.e. end of: do i=1,nofFracFiles
      end do

      close (72, iostat=errorMarker)


c     Write the savde fractures to DT Binary fracture file


c     Read the name of the file Where to put intersecting fractures
      write(6,*) 'Enter the name of the output file'
      read(5,*) outputFile
      write(6,*) outputFile



c     Open the file containing the paths to the scanline files
      open (74, file=outputFile, form='unformatted', IOSTAT=errorMarker)
      if (errorMarker.ne.0) then
        write (6, *) 'Could not open input file, aborting.'
        goto 990
      endif

      ns=6
      nv=8
c     Write the file header
```

```
      write (74) '#FR#V310'

      do j=1,nofSL
         do i=1,nofHits(j)
            write (74) ns, nv, .false. , fracSetMatrix(j,i),
     &      fracMatrix(j, i, 1),
     &      fracMatrix(j, i, 2),
     &      fracMatrix(j, i, 3),
     &      fracMatrix(j, i, 4),
     &      fracMatrix(j, i, 5),
     &      fracMatrix(j, i, 6),
     &      fracMatrix(j, i, 7),
     &      fracMatrix(j, i, 8),
     &      fracMatrix(j, i, 9),
     &      fracMatrix(j, i, 10),
     &      fracMatrix(j, i, 11),
     &      fracMatrix(j, i, 12),
     &      fracMatrix(j, i, 13),
     &      fracMatrix(j, i, 14),
     &      fracMatrix(j, i, 15),
     &      fracMatrix(j, i, 16),
     &      fracMatrix(j, i, 17),
     &      fracMatrix(j, i, 18),
     &      fracMatrix(j, i, 19),
     &      fracMatrix(j, i, 20),
     &      fracMatrix(j, i, 21),
     &      fracMatrix(j, i, 22),
     &      fracMatrix(j, i, 23),
     &      fracMatrix(j, i, 24),
     &      fracMatrix(j, i, 25),
     &      fracMatrix(j, i, 26),
     &      fracMatrix(j, i, 27),
     &      fracMatrix(j, i, 28),
     &      fracMatrix(j, i, 29),
     &      fracMatrix(j, i, 30),
     &      fracMatrix(j, i, 31),
     &      fracMatrix(j, i, 32),
     &      fracMatrix(j, i, 33),
     &      fracMatrix(j, i, 34),
     &      fracMatrix(j, i, 35),
     &      fracMatrix(j, i, 36),
     &      fracMatrix(j, i, 37),
     &      fracMatrix(j, i, 38),
     &      fracMatrix(j, i, 39),
     &      fracMatrix(j, i, 40),
     &      fracMatrix(j, i, 41),
     &      fracMatrix(j, i, 42),
     &      fracMatrix(j, i, 43),
     &      fracMatrix(j, i, 44),
     &      fracMatrix(j, i, 45),
     &      fracMatrix(j, i, 46),
     &      fracMatrix(j, i, 47),
     &      fracMatrix(j, i, 48),
     &      fracMatrix(j, i, 49),
     &      fracMatrix(j, i, 50),
     &      fracMatrix(j, i, 51),
     &      fracMatrix(j, i, 52),
     &      fracMatrix(j, i, 53),
     &      fracMatrix(j, i, 54),
     &      fracMatrix(j, i, 55),
     &      fracMatrix(j, i, 56),
     &      fracMatrix(j, i, 57),
     &      fracMatrix(j, i, 58),
     &      fracMatrix(j, i, 59)
        end do
      end do

      close (74, iostat=errorMarker)

990   continue

      end
```

A.2 Evaluation of realisations (Visual Basic .Net 2003)

```
Option Explicit On
Module Module1


    Sub Main()
        Const ForReading = 1, ForWriting = 2, ForAppending = 3
        Const TristateUseDefault = -2, TristateTrue = -1, TristateFalse = 0

        Dim changed As Boolean

        Dim i, j, k, nofFraxInFile, nofData, tmpND, nofReal As Integer
        Dim OutFile As String
        Dim inputFile As String
        Dim errorMessage As String
        Dim realNo As String
        Dim model As String
        Dim sDummy As String
        Dim fsInp, fInp, tsInp, fsOut, fOut, tsOut
        Dim response As MsgBoxResult
        Dim a, b, s, sN01, pi As Double
        Dim tSumCorr, tSumSemi As Double

        Dim readMatrix(6, 100) As Double
        Dim sortMatrix(6) As Double
        Dim dataMatrix(7, 100000) As Double
        Dim tmpMatrix(7, 100000) As Double

        pi = Math.Atan(1) * 4


        'Choose the appropriate model
        'model = "HRD_C"
        'model = "HRD_W"
        'model = "HRD_EW007"
        model = "FFM01"
        nofReal = 1000

        'Create the read and print objects
        fsInp = CreateObject("Scripting.FileSystemObject")
        fsOut = CreateObject("Scripting.FileSystemObject")

        For i = 1 To nofReal

            If i < 10 Then
                realNo = "000" & Trim(Str(i))
            ElseIf i < 100 Then
                realNo = "00" & Trim(Str(i))
            ElseIf i < 1000 Then
                realNo = "0" & Trim(Str(i))
            Else
                realNo = Trim(Str(i))
            End If

            inputFile = "xIng_data" & realNo & ".txt"

            fInp = fsInp.GetFile(inputFile)
            tsInp = fInp.OpenAsTextStream(ForReading, TristateUseDefault)

            'Read Header in file
            sDummy = tsInp.ReadLine
            j = 0
            While tsInp.AtEndOfStream <> True
                j = j + 1

                'Read Correlated Transmissivity
                sDummy = tsInp.ReadLine
                k = 1
                While Mid(sDummy, k, 1) <> ","
                    k = k + 1
                End While
                readMatrix(1, j) = Mid(sDummy, 1, k - 1)

                'Read set
```

```
                sDummy = Mid(sDummy, k + 1)
                k = 1
                While Mid(sDummy, k, 1) <> ","
                    k = k + 1
                End While
                readMatrix(2, j) = Mid(sDummy, 1, k - 1)

                'Read radius
                sDummy = Mid(sDummy, k + 1)
                k = 1
                While Mid(sDummy, k, 1) <> ","
                    k = k + 1
                End While
                readMatrix(3, j) = Mid(sDummy, 1, k - 1)

                'Read Crosspoint
                sDummy = Mid(sDummy, k + 1)
                k = 1
                While Mid(sDummy, k, 1) <> ","
                    k = k + 1
                End While
                readMatrix(4, j) = Mid(sDummy, 1, k - 1)

                'Read Strike
                sDummy = Mid(sDummy, k + 1)
                k = 1
                While Mid(sDummy, k, 1) <> ","
                    k = k + 1
                End While
                readMatrix(5, j) = Mid(sDummy, 1, k - 1)

                'Read Dip
                readMatrix(6, j) = Mid(sDummy, k + 1)
            End While

            nofFraxInFile = j

            'Sort data on length, using bubble sort
            changed = True
            While changed
                changed = False
                For j = 1 To nofFraxInFile - 1
                    If readMatrix(4, j) > readMatrix(4, j + 1) Then
                        For k = 1 To 6
                            sortMatrix(k) = readMatrix(k, j + 1)
                            readMatrix(k, j + 1) = readMatrix(k, j)
                            readMatrix(k, j) = sortMatrix(k)
                        Next
                        changed = True
                    End If
                Next j
            End While


            'Save Data in the Major Matrix
            For j = 1 To nofFraxInFile
                For k = 1 To 3
                    dataMatrix(k, j + nofData) = readMatrix(k, j)
                Next k
                '##OBSERVE THAT THE LENGTH DATA IS ADDED 100 FOR EACH FILE SO THAT THE
REALISATION WILL BE A 100000m LONG HOLE
                dataMatrix(4, j + nofData) = readMatrix(4, j) + 100 * (i - 1)
                For k = 5 To 6
                    dataMatrix(k, j + nofData) = readMatrix(k, j)
                Next k
            Next j
            nofData = nofData + nofFraxInFile
        Next i



        'Calculate the semicorrelated Transmissivities
        Randomize(1)
        For i = 1 To nofData

            If model = "HRD_C" Then
                If dataMatrix(2, i) = 1 Then
                    a = 0.0000005
                    b = 0.5
```

```
            s = 0.5
        ElseIf dataMatrix(2, i) = 2 Then
            a = 0.00000002
            b = 0.6
            s = 0.4
        ElseIf dataMatrix(2, i) = 3 Then
            a = 0.00000001
            b = 0.4
            s = 0.4
        ElseIf dataMatrix(2, i) = 4 Then
            a = 0.0000003
            b = 0.6
            s = 0.6
        End If
    ElseIf model = "HRD_W" Then
        If dataMatrix(2, i) = 1 Then
            a = 0.000000003
            b = 0.6
            s = 0.5
        ElseIf dataMatrix(2, i) = 2 Then
            a = 0.00000003
            b = 0.6
            s = 0.5
        ElseIf dataMatrix(2, i) = 3 Then
            a = 0.00000003
            b = 0.4
            s = 0.4
        ElseIf dataMatrix(2, i) = 4 Then
            a = 0.0000005
            b = 0.4
            s = 1.0
        End If
    ElseIf model = "HRD_EW007" Then
        If dataMatrix(2, i) = 1 Then
            a = 0.00000003
            b = 0.4
            s = 0.4
        ElseIf dataMatrix(2, i) = 2 Then
            a = 0.0000001
            b = 0.3
            s = 0.3
        ElseIf dataMatrix(2, i) = 3 Then
            a = 0.0000003
            b = 0.4
            s = 0.4
        ElseIf dataMatrix(2, i) = 4 Then
            a = 0.00000003
            b = 0.6
            s = 0.4
        End If
    ElseIf model = "FFM01" Then
        a = 0.00000003
        b = 0.4
        s = 0.4
    End If
    sN01 = s * (-2 * Math.Log(Rnd)) ^ 0.5 * Math.Sin(2 * pi * Rnd())
    dataMatrix(7, i) = 10 ^ (Math.Log10(a * dataMatrix(3, i) ^ b) + sN01)
Next i


'WRITE ALL THE OUTPUTFILES

'Write all fractures
OutFile = "L_All_" & model & ".txt"
'create the file string object for output file
fsOut.CreateTextFile(OutFile)
fOut = fsOut.GetFile(OutFile)
tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
For i = 1 To nofData - 1
    tsOut.WriteLine(dataMatrix(4, i + 1) - dataMatrix(4, i))
Next i
tsOut.Close()
'Write orientation for all fractures
OutFile = "Ori_All_" & model & ".txt"
'create the file string object for output file
fsOut.CreateTextFile(OutFile)
fOut = fsOut.GetFile(OutFile)
tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
For i = 1 To nofData
```

```
            tsOut.WriteLine(dataMatrix(2, i) & Chr(9) & dataMatrix(3, i) & Chr(9) &
dataMatrix(5, i) & Chr(9) & dataMatrix(6, i))
        Next i
        tsOut.Close()


        'WRITE THE CORRELATED FRACTURE DATA

        'Find all Correlated fractures with T>4e-9
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(1, i) > 0.000000004 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write  Correlated fractures with T>4e-9
        OutFile = "L_4e-9_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write Correlated fractures orientation with T>4e-9
        OutFile = "Ori_4e-9_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all  Correlated fractures with T>3e-8
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(1, i) > 0.00000003 Then
                tmpND = tmpND + 1
                For j = 1 To 6
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write Correlated fractures with T>3e-8
        OutFile = "L_3e-8_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write Correlated fractures orientation with 3>4e-8
        OutFile = "Ori_3e-8_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all Correlated fractures with T>2e-7
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(1, i) > 0.0000002 Then
                tmpND = tmpND + 1
```

```
                For j = 1 To 6
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write Correlated fractures with T>2e-7
        OutFile = "L_2e-7_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write Correlated fractures orientation with T>2e-7
        OutFile = "Ori_2e-7_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all Correlated fractures with T>5e-7
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(1, i) > 0.0000005 Then
                tmpND = tmpND + 1
                For j = 1 To 6
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write Correlated fractures with T>5e-7
        OutFile = "L_5e-7_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write Correlated fractures orientation with T>4e-9
        OutFile = "Ori_5e-7_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all Correlated fractures with T>1e-6
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(1, i) > 0.000001 Then
                tmpND = tmpND + 1
                For j = 1 To 6
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write Correlated fractures with T>1e-6
        OutFile = "L_1e-6_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
```

```
        Next i
        tsOut.Close()
        'Write Correlated fractures orientation with T>1e-6
        OutFile = "Ori_1e-6_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'WRITE THE SEMI CORRELATED FRACTURE DATA

        'Find all semi Correlated fractures with T>4e-9
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(7, i) > 0.000000004 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write semi Correlated fractures with T>4e-9
        OutFile = "L_4e-9_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write semi Correlated fractures orientation with T>4e-9
        OutFile = "Ori_4e-9_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all semi Correlated fractures with T>3e-8
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(7, i) > 0.00000003 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write semi Correlated fractures with T>3e-8
        OutFile = "L_3e-8_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write semi Correlated fractures orientation with T>3e-8
        OutFile = "Ori_3e-8_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
```

```
        Next i
        tsOut.Close()


        'Find all semi Correlated fractures with T>2e-7
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(7, i) > 0.0000002 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write semi Correlated fractures with T>2e-7
        OutFile = "L_2e-7_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write semi Correlated fractures orientation with T>2e-7
        OutFile = "Ori_2e-7_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()

        'Find all semi Correlated fractures with T>5e-7
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(7, i) > 0.0000005 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
        'Write semi Correlated fractures with T>5e-7
        OutFile = "L_5e-7_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write semi Correlated fractures orientation with T>5e-7
        OutFile = "Ori_5e-7_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Find all semi Correlated fractures with T>1e-6
        tmpND = 0
        For i = 1 To nofData
            If dataMatrix(7, i) > 0.000001 Then
                tmpND = tmpND + 1
                For j = 1 To 7
                    tmpMatrix(j, tmpND) = dataMatrix(j, i)
                Next j
            End If
        Next
```

```
        'Write semi Correlated fractures with T>1e-6
        OutFile = "L_1e-6_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND - 1
            tsOut.WriteLine(tmpMatrix(4, i + 1) - tmpMatrix(4, i))
        Next i
        tsOut.Close()
        'Write semi Correlated fractures orientation with T>1e-6
        OutFile = "Ori_1e-6_semi_corr_" & model & ".txt"
        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To tmpND
            tsOut.WriteLine(tmpMatrix(2, i) & Chr(9) & tmpMatrix(3, i) & Chr(9) & tmpMatrix(5,
i) & Chr(9) & tmpMatrix(6, i))
        Next i
        tsOut.Close()


        'Write All data in file
        OutFile = "All_data" & model & ".txt"

        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        For i = 1 To nofData
            tsOut.WriteLine(dataMatrix(1, i) & Chr(9) & dataMatrix(2, i) & Chr(9) &
dataMatrix(3, i) & Chr(9) & dataMatrix(4, i) & Chr(9) & dataMatrix(5, i) & Chr(9) &
dataMatrix(6, i) & Chr(9) & dataMatrix(7, i))
        Next i
        tsOut.Close()


        'Write sum of Transmissivity for 100m pieces
        OutFile = "Tsum_100m_" & model & ".txt"

        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        tSumCorr = 0
        tSumSemi = 0
        j = 1
        For i = 1 To nofData
            If dataMatrix(4, i) < j * 100 Then
                tSumCorr = tSumCorr + dataMatrix(1, i)
                tSumSemi = tSumSemi + dataMatrix(7, i)
            Else
                tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
                tSumCorr = dataMatrix(1, i)
                tSumSemi = dataMatrix(7, i)
                j = j + 1
                While dataMatrix(4, i) > j * 100
                    j = j + 1
                    tsOut.WriteLine("0" & Chr(9) & "0")
                End While
            End If
        Next i
        tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
        While j < 1000
            tsOut.WriteLine("0" & Chr(9) & "0")
            j = j + 1
        End While
        tsOut.Close()

        'Write sum of Transmissivity for 20m pieces
        OutFile = "Tsum_20m_" & model & ".txt"

        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        tSumCorr = 0
        tSumSemi = 0
```

```
        j = 1
        For i = 1 To nofData
            If dataMatrix(4, i) < j * 20 Then
                tSumCorr = tSumCorr + dataMatrix(1, i)
                tSumSemi = tSumSemi + dataMatrix(7, i)
            Else
                tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
                tSumCorr = dataMatrix(1, i)
                tSumSemi = dataMatrix(7, i)
                j = j + 1
                While dataMatrix(4, i) > j * 20
                    j = j + 1
                    tsOut.WriteLine("0" & Chr(9) & "0")
                End While
            End If
        Next i
        tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
        While j < 5000
            tsOut.WriteLine("0" & Chr(9) & "0")
            j = j + 1
        End While

        tsOut.Close()


        'Write sum of Transmissivity for 8m pieces
        OutFile = "Tsum_8m_" & model & ".txt"

        'create the file string object for output file
        fsOut.CreateTextFile(OutFile)
        fOut = fsOut.GetFile(OutFile)
        tsOut = fOut.OpenAsTextStream(ForWriting, TristateUseDefault)
        tSumCorr = 0
        tSumSemi = 0
        j = 1
        For i = 1 To nofData
            If dataMatrix(4, i) < j * 8 Then
                tSumCorr = tSumCorr + dataMatrix(1, i)
                tSumSemi = tSumSemi + dataMatrix(7, i)
            Else
                tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
                tSumCorr = dataMatrix(1, i)
                tSumSemi = dataMatrix(7, i)
                j = j + 1
                While dataMatrix(4, i) > j * 8
                    j = j + 1
                    tsOut.WriteLine("0" & Chr(9) & "0")
                End While
            End If
        Next i
        tsOut.WriteLine(tSumCorr & Chr(9) & tSumSemi)
        While j < 12500
            tsOut.WriteLine("0" & Chr(9) & "0")
            j = j + 1
        End While

        tsOut.Close()


errorTrap:
    End Sub

End Module
```

A.3 Generation of indata files (Visual Basic .Net 2003)

```
Option Explicit On
Module Module1


    Sub Main()
        Dim i, j, k, nofRealisations As Integer

        Const ForReading = 1, ForWriting = 2, ForAppending = 3
        Const TristateUseDefault = -2, TristateTrue = -1, TristateFalse = 0

        Dim fsBat, fBat, tsBat, fsCIF, fCIF, tsCIF, fsTxt, fTxt, tsTxt
        Dim batFileName, currCIFfileName, errorMessage As String
        Dim response As MsgBoxResult
        Dim xExt, yExt, zExt As Single
        Dim fractureSizes(5) As Single
        Dim boxSizes(4) As Single
        Dim fractureGenData(5, 7) As Single 'expid, idref, lambda1, lambda2, lambda3, tconstA,
tconstB)
        Dim fractureDomain As String
        Dim realNo As String


        nofRealisations = 1000
        'batFileName = "run1000_HRD_C.bat"
        'batFileName = "run1000_HRD_W.bat"
        'batFileName = "run1000_HRD_EW007.bat"
        batFileName = "run1000_FFM01.bat"

        'extension of sampling tunnel
        'xExt = 100
        xExt = 0
        yExt = 0
        'zExt = 0
        zExt = 100

        '## ROTATED -40 grader
        'fractureDomain = "HRD_C"
        ''Fracture Set 1
        'fractureGenData(1, 1) = -2.8        'expid
        'fractureGenData(1, 2) = 0.03513664 'idref
        'fractureGenData(1, 3) = 8.67816     'lambda1
        'fractureGenData(1, 4) = -4.06515    'lambda2
        'fractureGenData(1, 5) = -0.56934    'lambda3
        'fractureGenData(1, 6) = 0.000000504799 'tconsta
        'fractureGenData(1, 7) = 0.7 'tconstb
        ''Fracture Set 2
        'fractureGenData(2, 1) = -2.5 'expid
        'fractureGenData(2, 2) = 0.09681898 'idref
        'fractureGenData(2, 3) = 3.28623 'lambda1
        'fractureGenData(2, 4) = -11.5364  'lambda2
        'fractureGenData(2, 5) = -0.33506 'lambda3
        'fractureGenData(2, 6) = 0.000000113085 'tconsta
        'fractureGenData(2, 7) = 0.9 'tconstb
        ''Fracture Set 3
        'fractureGenData(3, 1) = -2.9 'expid
        'fractureGenData(3, 2) = 0.03731982 'idref
        'fractureGenData(3, 3) = -5.92832 'lambda1
        'fractureGenData(3, 4) = -4.93929 'lambda2
        'fractureGenData(3, 5) = -1.13945 'lambda3
        'fractureGenData(3, 6) = 0.0000000751126 'tconsta
        'fractureGenData(3, 7) = 0.5 'tconstb
        ''Fracture Set 4
        'fractureGenData(4, 1) = -2.9 'expid
        'fractureGenData(4, 2) = 0.04605424 'idref
        'fractureGenData(4, 3) = 0.12163 'lambda1
        'fractureGenData(4, 4) = 1.10175  'lambda2
        'fractureGenData(4, 5) = -11.9487 'lambda3
        'fractureGenData(4, 6) = 0.00000565423 'tconsta
        'fractureGenData(4, 7) = 0.9 'tconstb

        'fractureDomain = "HRD_W"
        ''Fracture Set 1
        'fractureGenData(1, 1) = -2.8        'expid
        'fractureGenData(1, 2) = 0.01571902 'idref
        'fractureGenData(1, 3) = -12.98753  'lambda1
        'fractureGenData(1, 4) = 7.49836     'lambda2
```

```
'fractureGenData(1, 5) = -0.31414   'lambda3
'fractureGenData(1, 6) = 0.0000000112423 'tconsta
'fractureGenData(1, 7) = 0.6 'tconstb
''Fracture Set 2
'fractureGenData(2, 1) = -2.55 'expid
'fractureGenData(2, 2) = 0.04136992 'idref
'fractureGenData(2, 3) = 2.09694 'lambda1
'fractureGenData(2, 4) = -10.68821  'lambda2
'fractureGenData(2, 5) = -0.41843 'lambda3
'fractureGenData(2, 6) = 0.000000336533 'tconsta
'fractureGenData(2, 7) = 0.7 'tconstb
''Fracture Set 3
'fractureGenData(3, 1) = -2.55 'expid
'fractureGenData(3, 2) = 0.03760902 'idref
'fractureGenData(3, 3) = -8.95992 'lambda1
'fractureGenData(3, 4) = -6.80095 'lambda2
'fractureGenData(3, 5) = -2.39098 'lambda3
'fractureGenData(3, 6) = 0.000000112423 'tconsta
'fractureGenData(3, 7) = 0.6 'tconstb
''Fracture Set 4
'fractureGenData(4, 1) = -2.65 'expid
'fractureGenData(4, 2) = 0.04285236 'idref
'fractureGenData(4, 3) = -0.92564 'lambda1
'fractureGenData(4, 4) = -0.59882  'lambda2
'fractureGenData(4, 5) = -11.04512 'lambda3
'fractureGenData(4, 6) = 0.0000151667 'tconsta
'fractureGenData(4, 7) = 1.2 'tconstb

'fractureDomain = "HRD_EW007"
''Fracture Set 1
'fractureGenData(1, 1) = -2.95       'expid
'fractureGenData(1, 2) = 0.05053132 'idref
'fractureGenData(1, 3) = 8.99138  'lambda1
'fractureGenData(1, 4) = -5.79455    'lambda2
'fractureGenData(1, 5) = -0.26142   'lambda3
'fractureGenData(1, 6) = 0.000000150554 'tconsta
'fractureGenData(1, 7) = 0.4 'tconstb
''Fracture Set 2
'fractureGenData(2, 1) = -2.5 'expid
'fractureGenData(2, 2) = 0.18709614 'idref
'fractureGenData(2, 3) = -4.1616  'lambda1
'fractureGenData(2, 4) = 15.86309  'lambda2
'fractureGenData(2, 5) = -0.05725 'lambda3
'fractureGenData(2, 6) = 0.000000335295 'tconsta
'fractureGenData(2, 7) = 0.3 'tconstb
''Fracture Set 3
'fractureGenData(3, 1) = -2.95 'expid
'fractureGenData(3, 2) = 0.04686963 'idref
'fractureGenData(3, 3) = 6.616   'lambda1
'fractureGenData(3, 4) = 5.77151 'lambda2
'fractureGenData(3, 5) = -0.59853 'lambda3
'fractureGenData(3, 6) = 0.00000150554 'tconsta
'fractureGenData(3, 7) = 0.4 'tconstb
''Fracture Set 4
'fractureGenData(4, 1) = -2.95 'expid
'fractureGenData(4, 2) = 0.06737509 'idref
'fractureGenData(4, 3) = 1.45035 'lambda1
'fractureGenData(4, 4) = -0.22193  'lambda2
'fractureGenData(4, 5) = -9.58839 'lambda3
'fractureGenData(4, 6) = 0.000000337269 'tconsta
'fractureGenData(4, 7) = 0.6 'tconstb

'## ROTATED -50 grader
fractureDomain = "FFM01"
'Fracture Set 1 "NS"
fractureGenData(1, 1) = -2.5       'expid
fractureGenData(1, 2) = 0.012298627 'idref
fractureGenData(1, 3) = -15.71407 'lambda1
fractureGenData(1, 4) = -8.35532 'lambda2
fractureGenData(1, 5) = -0.31065 'lambda3
fractureGenData(1, 6) = 0.0000000101554 'tconsta
fractureGenData(1, 7) = 1.0 'tconstb
'Fracture Set 2 "NE"
fractureGenData(2, 1) = -2.75 'expid
fractureGenData(2, 2) = 0.016174872 'idref
fractureGenData(2, 3) = -14.213 'lambda1
fractureGenData(2, 4) = 1.49385 'lambda2
fractureGenData(2, 5) = -0.49906 'lambda3
fractureGenData(2, 6) = 0.0000000101554 'tconsta
```

```
fractureGenData(2, 7) = 1.0 'tconstb
'Fracture Set 3 "NW"
fractureGenData(3, 1) = -3.1 'expid
fractureGenData(3, 2) = 0.005544014 'idref
fractureGenData(3, 3) = 2.22779 'lambda1
fractureGenData(3, 4) = 12.63443 'lambda2
fractureGenData(3, 5) = -1.34842 'lambda3
fractureGenData(3, 6) = 0.0000000101554 'tconsta
fractureGenData(3, 7) = 1.0 'tconstb
'Fracture Set 4 "EW"
fractureGenData(4, 1) = -3.1 'expid
fractureGenData(4, 2) = 0.002206291 'idref
fractureGenData(4, 3) = -8.02518 'lambda1
fractureGenData(4, 4) = 11.46114 'lambda2
fractureGenData(4, 5) = -0.48859 'lambda3
fractureGenData(4, 6) = 0.0000000101554 'tconsta
fractureGenData(4, 7) = 1.0 'tconstb
'Fracture Set 5 "HZ"
fractureGenData(5, 1) = -2.38 'expid
fractureGenData(5, 2) = 0.01973405 'idref
fractureGenData(5, 3) = -0.74974 'lambda1
fractureGenData(5, 4) = 0.74974 'lambda2
fractureGenData(5, 5) = -15.16297 'lambda3
fractureGenData(5, 6) = 0.0000000101554 'tconsta
fractureGenData(5, 7) = 1.0 'tconstb


'## NO ROTATION
'fractureDomain = "HRD_C"
''Fracture Set 1
'fractureGenData(1, 1) = -2.8       'expid
'fractureGenData(1, 2) = 0.03513664 'idref
'fractureGenData(1, 3) = 4.03483    'lambda1
'fractureGenData(1, 4) = -8.6923    'lambda2
'fractureGenData(1, 5) = -0.56934   'lambda3
'fractureGenData(1, 6) = 0.000000504799 'tconsta
'fractureGenData(1, 7) = 0.7 'tconstb
''Fracture Set 2
'fractureGenData(2, 1) = -2.5 'expid
'fractureGenData(2, 2) = 0.09681898 'idref
'fractureGenData(2, 3) = -4.89806 'lambda1
'fractureGenData(2, 4) = -10.94974  'lambda2
'fractureGenData(2, 5) = -0.33506 'lambda3
'fractureGenData(2, 6) = 0.000000113085 'tconsta
'fractureGenData(2, 7) = 0.9 'tconstb
''Fracture Set 3
'fractureGenData(3, 1) = -2.9 'expid
'fractureGenData(3, 2) = 0.03731982 'idref
'fractureGenData(3, 3) = -7.71628 'lambda1
'fractureGenData(3, 4) = 0.02693 'lambda2
'fractureGenData(3, 5) = -1.13945 'lambda3
'fractureGenData(3, 6) = 0.0000000751126 'tconsta
'fractureGenData(3, 7) = 0.5 'tconstb
''Fracture Set 4
'fractureGenData(4, 1) = -2.9 'expid
'fractureGenData(4, 2) = 0.04605424 'idref
'fractureGenData(4, 3) = 0.80137 'lambda1
'fractureGenData(4, 4) = 0.76581  'lambda2
'fractureGenData(4, 5) = -11.9487 'lambda3
'fractureGenData(4, 6) = 0.00000565423 'tconsta
'fractureGenData(4, 7) = 0.9 'tconstb

'fractureDomain = "HRD_W"
''Fracture Set 1
'fractureGenData(1, 1) = -2.8       'expid
'fractureGenData(1, 2) = 0.01571902 'idref
'fractureGenData(1, 3) = -5.12918  'lambda1
'fractureGenData(1, 4) = 14.0923     'lambda2
'fractureGenData(1, 5) = -0.31414   'lambda3
'fractureGenData(1, 6) = 0.0000000112423 'tconsta
'fractureGenData(1, 7) = 0.6 'tconstb
''Fracture Set 2
'fractureGenData(2, 1) = -2.55 'expid
'fractureGenData(2, 2) = 0.04136992 'idref
'fractureGenData(2, 3) = -5.2639  'lambda1
'fractureGenData(2, 4) = -9.53553  'lambda2
'fractureGenData(2, 5) = -0.41843 'lambda3
'fractureGenData(2, 6) = 0.000000336533 'tconsta
'fractureGenData(2, 7) = 0.7 'tconstb
```

```
''Fracture Set 3
'fractureGenData(3, 1) = -2.55 'expid
'fractureGenData(3, 2) = 0.03760902 'idref
'fractureGenData(3, 3) = -11.23527 'lambda1
'fractureGenData(3, 4) = 0.5495  'lambda2
'fractureGenData(3, 5) = -2.39098 'lambda3
'fractureGenData(3, 6) = 0.000000112423 'tconsta
'fractureGenData(3, 7) = 0.6 'tconstb
''Fracture Set 4
'fractureGenData(4, 1) = -2.65 'expid
'fractureGenData(4, 2) = 0.04285236 'idref
'fractureGenData(4, 3) = -1.094   'lambda1
'fractureGenData(4, 4) = 0.13626  'lambda2
'fractureGenData(4, 5) = -11.04512 'lambda3
'fractureGenData(4, 6) = 0.0000151667 'tconsta
'fractureGenData(4, 7) = 1.2 'tconstb

'fractureDomain = "HRD_EW007"
''Fracture Set 1
'fractureGenData(1, 1) = -2.95       'expid
'fractureGenData(1, 2) = 0.05053132 'idref
'fractureGenData(1, 3) = 3.16313  'lambda1
'fractureGenData(1, 4) = -10.21843    'lambda2
'fractureGenData(1, 5) = -0.26142   'lambda3
'fractureGenData(1, 6) = 0.000000150554 'tconsta
'fractureGenData(1, 7) = 0.4 'tconstb
''Fracture Set 2
'fractureGenData(2, 1) = -2.5 'expid
'fractureGenData(2, 2) = 0.18709614 'idref
'fractureGenData(2, 3) = 7.00863  'lambda1
'fractureGenData(2, 4) = 14.82686  'lambda2
'fractureGenData(2, 5) = -0.05725 'lambda3
'fractureGenData(2, 6) = 0.000000335295 'tconsta
'fractureGenData(2, 7) = 0.3 'tconstb
''Fracture Set 3
'fractureGenData(3, 1) = -2.95 'expid
'fractureGenData(3, 2) = 0.04686963 'idref
'fractureGenData(3, 3) = 8.778     'lambda1
'fractureGenData(3, 4) = 0.16855 'lambda2
'fractureGenData(3, 5) = -0.59853 'lambda3
'fractureGenData(3, 6) = 0.00000150554 'tconsta
'fractureGenData(3, 7) = 0.4 'tconstb
''Fracture Set 4
'fractureGenData(4, 1) = -2.95 'expid
'fractureGenData(4, 2) = 0.06737509 'idref
'fractureGenData(4, 3) = 0.96837 'lambda1
'fractureGenData(4, 4) = -1.10228  'lambda2
'fractureGenData(4, 5) = -9.58839 'lambda3
'fractureGenData(4, 6) = 0.000000337269 'tconsta
'fractureGenData(4, 7) = 0.6 'tconstb

'fractureDomain = "FFM01"
''Fracture Set 1 "NS"
'fractureGenData(1, 1) = -2.5       'expid
'fractureGenData(1, 2) = 0.012298627 'idref
'fractureGenData(1, 3) = -16.50136 'lambda1
'fractureGenData(1, 4) = 6.66698 'lambda2
'fractureGenData(1, 5) = -0.31065 'lambda3
'fractureGenData(1, 6) = 0.0000000101554 'tconsta
'fractureGenData(1, 7) = 1.0 'tconstb
''Fracture Set 2 "NE"
'fractureGenData(2, 1) = -2.75 'expid
'fractureGenData(2, 2) = 0.016174872 'idref
'fractureGenData(2, 3) = -7.99159 'lambda1
'fractureGenData(2, 4) = 11.84802 'lambda2
'fractureGenData(2, 5) = -0.49906 'lambda3
'fractureGenData(2, 6) = 0.0000000101554 'tconsta
'fractureGenData(2, 7) = 1.0 'tconstb
''Fracture Set 3 "NW"
'fractureGenData(3, 1) = -3.1 'expid
'fractureGenData(3, 2) = 0.005544014 'idref
'fractureGenData(3, 3) = 11.11053 'lambda1
'fractureGenData(3, 4) = 6.41467 'lambda2
'fractureGenData(3, 5) = -1.34842 'lambda3
'fractureGenData(3, 6) = 0.0000000101554 'tconsta
'fractureGenData(3, 7) = 1.0 'tconstb
''Fracture Set 4 "EW"
'fractureGenData(4, 1) = -3.1 'expid
'fractureGenData(4, 2) = 0.002206291 'idref
```

```
        'fractureGenData(4, 3) = 3.62126 'lambda1
        'fractureGenData(4, 4) = 13.51472 'lambda2
        'fractureGenData(4, 5) = -0.48859 'lambda3
        'fractureGenData(4, 6) = 0.0000000101554 'tconsta
        'fractureGenData(4, 7) = 1.0 'tconstb
        ''Fracture Set 5 "HZ"
        'fractureGenData(5, 1) = -2.38 'expid
        'fractureGenData(5, 2) = 0.01973405 'idref
        'fractureGenData(5, 3) = 0.09241 'lambda1
        'fractureGenData(5, 4) = 1.05626 'lambda2
        'fractureGenData(5, 5) = -15.16297 'lambda3
        'fractureGenData(5, 6) = 0.0000000101554 'tconsta
        'fractureGenData(5, 7) = 1.0 'tconstb


        fractureSizes(5) = 0.5
        fractureSizes(4) = 1
        fractureSizes(3) = 10
        fractureSizes(2) = 100
        fractureSizes(1) = 1000

        boxSizes(4) = 3
        boxSizes(3) = 30
        boxSizes(2) = 300
        boxSizes(1) = 3000


        'Create the file string objects
        fsBat = CreateObject("Scripting.FileSystemObject")
        fsTxt = CreateObject("Scripting.FileSystemObject")
        fsCIF = CreateObject("Scripting.FileSystemObject")

        If fsBat.FileExists(batFileName) Then
            response = MsgBox("The file already exsists, would you like to replace it?",
vbYesNo, "Existing file")
            If response = vbNo Then
                errorMessage = "The old file is kept"
                GoTo errorTrap
            End If
        End If

        'GENERATE THE BAT FILE COMMANDS
        'create the file string object
        fsBat.CreateTextFile(batFileName)
        fBat = fsBat.GetFile(batFileName)
        tsBat = fBat.OpenAsTextStream(ForWriting, TristateUseDefault)

        tsBat.Writeline("rem *****************************************")
        tsBat.Writeline("rem **********      " & fractureDomain & "      **********")
        tsBat.Writeline("rem ********** " & nofRealisations & " Realisations  **********")
        tsBat.Writeline("rem *****************************************")


        'GENERATE THE SCAN LINE FILE
        fsTxt.CreateTextFile("xAlign100m.txt")
        fTxt = fsTxt.GetFile("xAlign100m.txt")
        tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

        tsTxt.Writeline("2")
        tsTxt.Writeline(-xExt / 2 & "," & -yExt / 2 & "," & -zExt / 2)
        tsTxt.Writeline(xExt / 2 & "," & yExt / 2 & "," & zExt / 2)

        tsTxt.Close()

        'GENERATE THE SCAN LINES INPUT
        fsTxt.CreateTextFile("scanLineFiles.txt")
        fTxt = fsTxt.GetFile("scanLineFiles.txt")
        tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

        tsTxt.Writeline("1")
        tsTxt.Writeline("xAlign100m.txt")

        tsTxt.Close()


        For i = 1 To nofRealisations
            If i < 10 Then
                realNo = "000" & Trim(Str(i))
            ElseIf i < 100 Then
```

```
    realNo = "00" & Trim(Str(i))
ElseIf i < 1000 Then
    realNo = "0" & Trim(Str(i))
Else
    realNo = Trim(Str(i))
End If

'GENERATE THE BAT FILE COMMANDS
tsBat.Writeline("")
tsBat.Writeline("rem *****************************************")
tsBat.Writeline("rem ********** Realisation: " & Trim(Str(i)) & "       **********")
tsBat.Writeline("rem *****************************************")


'GENERATE THE SCAN LINE INPUT FILES
fsTxt.CreateTextFile("scanLine_input_" & realNo & ".txt")
fTxt = fsTxt.GetFile("scanLine_input_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("scanLineFiles.txt")
tsTxt.Writeline("fracFiles_" & realNo & ".txt")
tsTxt.Writeline("xIng_" & realNo & ".dat")

tsTxt.Close()


'GENERATE THE FRACFILES INPUT FILES
fsTxt.CreateTextFile("fracFiles_" & realNo & ".txt")
fTxt = fsTxt.GetFile("fracFiles_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("1")
tsTxt.Writeline("saved_" & realNo & ".dat")

tsTxt.Close()

'GENERATE THE DISCARDED SCAN LINE INPUT FILES
fsTxt.CreateTextFile("scanLine_input_disc_" & realNo & ".txt")
fTxt = fsTxt.GetFile("scanLine_input_disc_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("scanLineFiles.txt")
tsTxt.Writeline("disc_frax_" & realNo & ".txt")
tsTxt.Writeline("Not_xIng_" & realNo & ".dat")

tsTxt.Close()

'GENERATE THE DISCARDED FRACFILES INPUT FILES
fsTxt.CreateTextFile("disc_frax_" & realNo & ".txt")
fTxt = fsTxt.GetFile("disc_frax_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("1")
tsTxt.Writeline("discarded_" & realNo & ".dat")

tsTxt.Close()

'GENERATE THE INPUT FILE TO VizNsave FOR xIng FRACTURES
fsTxt.CreateTextFile("input_xIng_" & realNo & ".txt")
fTxt = fsTxt.GetFile("input_xIng_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("xIng_" & realNo & ".dat")
tsTxt.Writeline("xIng_" & realNo & ".plt")
tsTxt.Writeline("xIng_data" & realNo & ".txt")

tsTxt.Close()

'GENERATE THE INPUT FILE TO VizNsave FOR Not Connected xIng FRACTURES
fsTxt.CreateTextFile("input_disc_" & realNo & ".txt")
fTxt = fsTxt.GetFile("input_disc_" & realNo & ".txt")
tsTxt = fTxt.OpenAsTextStream(ForWriting, TristateUseDefault)

tsTxt.Writeline("Not_xIng_" & realNo & ".dat")
tsTxt.Writeline("Not_xIng_" & realNo & ".plt")
tsTxt.Writeline("Not_xIng_data" & realNo & ".txt")

tsTxt.Close()
```

```
            'GENERATE THE CIF FILES FOR SORTING FRACTURES
            currCIFfileName = "cif_SORT_SER_PFL_" & realNo & ".xml"
            fsCIF.CreateTextFile(currCIFfileName)
            fCIF = fsBat.GetFile(currCIFfileName)
            tsCIF = fCIF.OpenAsTextStream(ForWriting, TristateUseDefault)

            'Write the file
            tsCIF.WriteLine("<cif>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("<!--============================================================")
            tsCIF.WriteLine("============== Project: SER PFL-statistics     ================")
            tsCIF.WriteLine("============== Task:    Sort Farctures         ================")
            tsCIF.WriteLine("============== Case:    Domain: " & fractureDomain & "
================")
            tsCIF.WriteLine("==============          box   : 1-4            ================")
            tsCIF.WriteLine("==============          Lmax  : " & Format("####.##",
fractureSizes(1)) & "         ================")
            tsCIF.WriteLine("==============          Lmin  : " & Format("####.##",
fractureSizes(5)) & "         ================")
            tsCIF.WriteLine("==============          seed  : " & Format("0.0000", i / 10000) & "
================")
            tsCIF.WriteLine("============== Date:   " & Today() & "
================")
            tsCIF.WriteLine("============== Author:  SER_PFL_MakeBatNfiles.exe
================")
            tsCIF.WriteLine("============================================================-->")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("<!--============================================================")
            tsCIF.WriteLine("=============  FRACTURE NETWORK  ==============================")
            tsCIF.WriteLine("============================================================-->")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("<fracgen>")
            tsCIF.WriteLine(" <seed>     " & i / 10000 & "      </seed>")
            tsCIF.WriteLine(" <wkdivh>    100. 100. 100.       </wkdivh>")
            tsCIF.WriteLine("</fracgen>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("<fraccmds>")
            tsCIF.WriteLine(" <clear>  listOfFrax  </clear>")
            tsCIF.WriteLine(" <readfile> Frax_1_" & realNo & ".dat list1  </readfile>")
            tsCIF.WriteLine(" <readfile> Frax_2_" & realNo & ".dat list2  </readfile>")
            tsCIF.WriteLine(" <readfile> Frax_3_" & realNo & ".dat list3  </readfile>")
            tsCIF.WriteLine(" <readfile> Frax_4_" & realNo & ".dat list4  </readfile>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine(" <assign>   list1 AllFrax  </assign>")
            tsCIF.WriteLine(" <move>     list2 AllFrax  </move>")
            tsCIF.WriteLine(" <move>     list3 AllFrax  </move>")
            tsCIF.WriteLine(" <move>     list4 AllFrax  </move>")
            tsCIF.WriteLine(" <sortisol> dummy AllFrax saved discarded </sortisol>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine(" <writefile>  saved  saved_" & realNo & ".dat </writefile>")
            tsCIF.WriteLine(" <writefile>  discarded  discarded_" & realNo & ".dat
</writefile>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("</fraccmds>")
            tsCIF.WriteLine("")
            tsCIF.WriteLine("</cif>")

            'Close the cif file for sorting fractures
            tsCIF.Close()

            For j = 4 To 1 Step -1

                'GENERATE THE CIF FILES FOR PRODUCING FRACTURES
                currCIFfileName = "cif_GEN_SER_PFL_" & Trim(Str(j)) & "_" & realNo & ".xml"
                fsCIF.CreateTextFile(currCIFfileName)
                fCIF = fsBat.GetFile(currCIFfileName)
                tsCIF = fCIF.OpenAsTextStream(ForWriting, TristateUseDefault)


                tsCIF.WriteLine("<cif>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!--
============================================================")
                tsCIF.WriteLine("============== Project: SER PFL-statistics
================")
                tsCIF.WriteLine("============== Task:    Generate Farctures
================")
```

```
                tsCIF.WriteLine("============== Case:    Domain: " & fractureDomain & " 
================")
                tsCIF.WriteLine("==============          box   : " & Trim(Str(j)) & " 
================")
                tsCIF.WriteLine("==============          Lmax  : " & Format("####.##", 
fractureSizes(j)) & "        ================")
                tsCIF.WriteLine("==============          Lmin  : " & Format("####.##", 
fractureSizes(j + 1)) & "        ================")
                tsCIF.WriteLine("==============          seed  : " & Format("0.0000", i / 10000) 
& "         ================")
                tsCIF.WriteLine("============== Date:   " & Today() & " 
================")
                tsCIF.WriteLine("============== Author:  SER_PFL_MakeBatNfiles.exe 
================")
                tsCIF.WriteLine("==============================================================--
>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!--
=============================================================")
                tsCIF.WriteLine("=============  INPUT-OUTPUT 
==================================")
                tsCIF.WriteLine("==============================================================--
>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!-- Tecplot -->")
                tsCIF.WriteLine("<!-- ******* -->")
                tsCIF.WriteLine("<tecplot>")
                tsCIF.WriteLine("<name>  grid       </name>")
                tsCIF.WriteLine("<loc>   boxRegion  </loc>")
                tsCIF.WriteLine("</tecplot>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!--
=============================================================")
                tsCIF.WriteLine("=============  GRID GENERATION 
===============================")
                tsCIF.WriteLine("==============================================================--
>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!-- ggn -->")
                tsCIF.WriteLine("<!-- *** -->")
                tsCIF.WriteLine("<gridgen>")
                tsCIF.WriteLine(" <span>     " & xExt + boxSizes(j) & " " & yExt + boxSizes(j) & 
" " & zExt + boxSizes(j) & "    </span>")
                tsCIF.WriteLine(" <origin>  " & -(xExt + boxSizes(j)) / 2 & " " & -(yExt + 
boxSizes(j)) / 2 & " " & -(zExt + boxSizes(j)) / 2 & " " & "    </origin>")
                tsCIF.WriteLine(" <tecplot>   grid                   </tecplot>")
                tsCIF.WriteLine(" <ggn>       grid1                  </ggn>")
                tsCIF.WriteLine(" <binary>    T                      </binary>")
                tsCIF.WriteLine("</gridgen>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<ggn>")
                tsCIF.WriteLine(" <name>      grid1                       </name>")
                tsCIF.WriteLine(" <nbcells>   100000                      </nbcells>")
                tsCIF.WriteLine(" <isotropy>  F                           </isotropy>")
                tsCIF.WriteLine(" <dxmax>     " & fractureSizes(j) / 2 & " 
</dxmax>")
                tsCIF.WriteLine(" <dymax>     " & fractureSizes(j) / 2 & " 
</dymax>")
                tsCIF.WriteLine(" <dzmax>     " & fractureSizes(j) / 2 & " 
</dzmax>")
                tsCIF.WriteLine("</ggn>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!--
=============================================================")
                tsCIF.WriteLine("=============  FRACTURE NETWORK 
==============================")
                tsCIF.WriteLine("==============================================================--
>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<fracgen>")
                tsCIF.WriteLine(" <seed>      " & i / 10000 & "      </seed>")
                tsCIF.WriteLine(" <wkdivh>    " & fractureSizes(j) / 2 & " " & fractureSizes(j) 
/ 2 & " " & fractureSizes(j) / 2 & "  </wkd>")
                tsCIF.WriteLine("</fracgen>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<fraccmds>")
                tsCIF.WriteLine(" <clear>  Frax_" & Trim(Str(j)) & "_" & realNo & "  </clear>")
                tsCIF.WriteLine(" <genran> Frax_" & Trim(Str(j)) & "_" & realNo & "  </genran>")
```

```
                tsCIF.WriteLine(" <writefile>  Frax_" & Trim(Str(j)) & "_" & realNo & "  Frax_"
& Trim(Str(j)) & "_" & realNo & ".dat </writefile>")
                tsCIF.WriteLine("</fraccmds>")

                'For k = 1 To 4
                For k = 1 To 5
                    tsCIF.WriteLine("")
                    tsCIF.WriteLine("<!-- Fracture set " & Trim(Str(k)) & " -->")
                    tsCIF.WriteLine("<ranf>")
                    tsCIF.WriteLine(" <lref>   1.                      </lref>")
                    tsCIF.WriteLine(" <expid>  " & Trim(Str(fractureGenData(k, 1))) & "
</expid>")
                    tsCIF.WriteLine(" <idref>  " & Trim(Str(fractureGenData(k, 2))) & "
</idref>")
                    tsCIF.WriteLine(" <sizes> " & fractureSizes(j + 1) & " " & fractureSizes(j)
& "                  </sizes>")
                    tsCIF.WriteLine(" <thknes>  0.001                      </thknes>")
                    tsCIF.WriteLine(" <lambda> " & Trim(Str(fractureGenData(k, 3))) & " " &
Trim(Str(fractureGenData(k, 4))) & " " & Trim(Str(fractureGenData(k, 5))) & "     </lambda>")
                    tsCIF.WriteLine("")
                    tsCIF.WriteLine(" <frevol> 1e-3                    </frevol>")
                    tsCIF.WriteLine(" <fwsurf>  2.                      </fwsurf>")
                    tsCIF.WriteLine(" <storat>  1e-6                    </storat>")
                    tsCIF.WriteLine(" <tconst>   " & Trim(Str(fractureGenData(k, 6))) & " " &
Trim(Str(fractureGenData(k, 7))) & " 1.E0 0. </tconst>")
                    tsCIF.WriteLine(" <diffus>  5e-12                   </diffus>")
                    tsCIF.WriteLine("</ranf>")

                Next
                tsCIF.WriteLine("")
                tsCIF.WriteLine("<!--
============================================================")
                tsCIF.WriteLine("=============  OBJECTS & LOCATIONS
===========================")
                tsCIF.WriteLine("=============================================================--
>")
                tsCIF.WriteLine("<loc>")
                tsCIF.WriteLine(" <name>  boxRegion                    </name>")
                tsCIF.WriteLine(" <patch> " & -(boxSizes(j) + xExt) / 2 & " " & (boxSizes(j) +
xExt) / 2 & " " & -(boxSizes(j) + yExt) / 2 & " " & (boxSizes(j) + yExt) / 2 & " " & -
(boxSizes(j) + zExt) / 2 & " " & (boxSizes(j) + zExt) / 2 & " </patch>")
                tsCIF.WriteLine("</loc>")
                tsCIF.WriteLine("")
                tsCIF.WriteLine("</cif>")

                tsCIF.Close()

                'GENERATE THE BAT FILE COMMANDS
                tsBat.Writeline("")
                tsBat.Writeline("Rem Run Fracture Generation")
                tsBat.Writeline("copy cif_GEN_SER_PFL_" & Trim(Str(j)) & "_" & realNo & ".xml
cif.xml")
                tsBat.Writeline("runggn")
                tsBat.Writeline("runfgn")



            Next j

            'GENERATE THE BAT FILE COMMANDS
            tsBat.Writeline("")
            tsBat.Writeline("Rem Run sorting")
            tsBat.Writeline("copy cif_SORT_SER_PFL_" & realNo & ".xml cif.xml")
            tsBat.Writeline("runfgn")
            tsBat.Writeline("")

            tsBat.Writeline("Rem Run Scanline")
            tsBat.Writeline("DT_ScanLine_sampling<scanLine_input_" & realNo & ".txt")
            tsBat.Writeline("DT_ScanLine_sampling<scanLine_input_disc_" & realNo & ".txt")
            tsBat.Writeline("")

            tsBat.Writeline("Rem Run VizNsave")
            tsBat.Writeline("VizNsaveSERdata<input_xIng_" & realNo & ".txt")
            tsBat.Writeline("VizNsaveSERdata<input_disc_" & realNo & ".txt")
            tsBat.Writeline("")

            tsBat.Writeline("Rem Delete Unneccesary files")
            tsBat.Writeline("del f*.dat")
            tsBat.Writeline("del disca*.dat")
```

```
        Next i

        tsBat.Close()

errorTrap:
    End Sub

End Module
```