Scale dependence of fluid flow and solute transport in fracture networks of tight geological formations

by

Oluwaseun Joseph Akomolafe

B.S., University of Ilorin, Nigeria, 2014

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Geology
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Approved by:

Major Professor
Dr. Behzad Ghanbarian

Copyright

Abstract

Understanding the effect of fracture network heterogeneity on flow and transport is of great importance because fractures are primary pathways, particularly in tight formations with low permeability values. Although numerous studies were conducted to investigate solute transport in fractured media, simultaneous investigation of the effect of scale on flow and transport in tight reservoirs is very limited. More specifically, we are not aware of any study that has addressed how the continuous time random walk (CTRW) model parameters vary with scale. In this study, we carried out extensive numerical simulations in sparse fracture networks of sizes L = 20, 35, and 50 m under two different fracture densities (p30 = 0.05 and 0.1) using the discrete fracture network approach. The fractures in the networks were elliptical in shape, whose radii followed the truncated power-law distribution with exponent α = 1.5, 2, and 2.5. We simulated fluid flow based on the Reynolds equation and solute transport using the particle tracking approach. The solute transport behavior was quantified by fitting the CTRW model to the simulated arrival time distributions averaged over at least twenty realizations. Results showed that as the exponent α increased, the permeability of the networks decreased. We found non-Fickian solute transport behavior, deduced from small β values, in all the fracture networks studied here. We demonstrated that although the value of permeability might have reached the REV value, solute transport parameters could still be scale-dependent. Our numerical analyses disclosed the scale dependence of the CTRW model parameters on the geometrical properties and topological properties of fracture networks, confirmed through our regression analyses.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I want to express my deepest gratitude to my advisor Dr. Behzad Ghanbarian who made this work possible. His invaluable supervision, support, and tutelage carried me through all the stages of writing this thesis and my graduate program. I am also grateful to my committee members, Dr. Matthew Kirk, and Dr. Jeffrey Hyman, for their wisdom, comments, and suggestions, which influenced my simulation methods and critiquing my results.

I would also like to thank the Department of Geology for financial support through the Graduate Teaching Assistantship/Graduate Research Assistantship award.

Finally, my profound gratitude goes to all my family members for their financial and emotional support and prayers. Thank you, and God bless you all.

Dedication

This thesis is dedicated to the glory of God, the beginning and the end.

Chapter 1 - Introduction

Tight geologic formations have received great attention, particularly in the past two decades, due to broad applications, such as hydrogen and carbon dioxide storage, radioactive waste materials disposal, new sources of fossil fuels, and geothermal energy exploitation (Liu, 2017; Yuan & Wood, 2018; Zoback & Kohli, 2019). Fluid flow and contaminant migration through such low-porosity geologic formations are mainly controlled by fractures, their aperture and length distributions as well as their interconnectivity (Faybishenko et al., 2015; Long & Witherspoon, 1985). The presence of fractures across scales spanning from a few microns to several kilometers in nature have been reported. Such heterogeneities spanning orders of magnitude lead to spatial variability as well as scale dependence in fracture networks and their physical and hydraulic characteristics (Davy et al., 2006; Forstner & Laubach, 2022; Gong & Rossen, 2017; Lei et al., 2015).

The literature on flow and transport in fracture networks is vast and extensive. Theories (Berkowitz et al., 2016; Sahimi, 2011), experiments (Chen et al., 2017; Detwiler et al., 2000), and simulations (Moreno et al., 1990; Roubinet et al., 2022) were widely applied to better understand physical and hydraulic properties of fractures and their networks. For instance, Neretnieks et al. (1982) experimentally studied radionuclide migration in a natural fracture in a granite core using reactive and non-reactive tracers. Those authors measured the breakthrough experimentally and found that for non-reactive tracers channeling may occur, while for reactive tracers diffusion into the porous matrix was significant. In another study, Grisak and Pickens (1980) injected a tracer solution containing Cl and Ca ions into a large cylindrical sample of fracture obtained in a fresh excavation. The sample contained two orthogonal sets of fractures parallel to the long axis of the cylinder. Breakthrough curves for both ions showed significant

retardation of the ions relative to the average water velocity in the cylindrical. They attributed the retardation to the effect of diffusion of solutes from the fracture into the matrix. Hull et al. (1987) constructed 2D networks of fracture and experimentally investigated dispersion in them. They also carried out random walk simulations in similar networks to test the validity of some assumptions that they made in their models.

The discrete fracture network (DFN) is one of the widely applied numerical approaches in the literature to study solutes (Cacas et al., 1990; Hyman, 2020; Kang et al., 2017, 2020). One advantage of DFN models over equivalent porous medium models, e.g., stochastic continuum, is their ability to represent a broader range of transport phenomena. These can be explicitly related to fracture network properties, e.g., densities, length, and fracture aperture (S. Painter & Cvetkovic, 2005; Raats, 1973). Using the DFN approach, Kang et al. (2019) studied the effects of geological stress on fluid flow and tracer transport in 2D fracture networks based on actual field measurements. They found that the interaction of fracture geometry, aperture heterogeneity, and geological stress results in the formation of tiny apertures, resulting in non-Fickian behavior of fluid flow and tracer transport through the system.

Through numerical simulation of flow and transport in fracture networks, Hyman et al. (2019) investigated large-scale particle motion and solute breakthrough in sparse three-dimensional discrete fracture networks characterized by power-law distributed fracture lengths. By using the stream-tube model and Bernoulli continuous-time random walk model to predict breakthrough, they found that particle motion in fracture networks cannot be characterized by a constant velocity between the inlet and control pane at which the breakthrough curve is detected. Instead, the network structure requires that frequent velocity transitions be made as particles move through the system.

In the literature, various theoretical models, such as the advection-dispersion equation (Nielsen & Biggar, 1962; Raats, 1973), mobile-immobile model (Coats & Smith, 1964; van Genuchten & Wagenet, 1989), fractional derivatives (Benson et al., 2000; Kelly et al., 2017), percolation theory (Ghanbarian-Alavijeh et al., 2012; Hunt et al., 2011; Hunt & Ghanbarian, 2016), continuous-time random walk (Berkowitz et al., 2006; Cortis & Berkowitz, 2004), and spatial Markov model (Kang et al., 2011; T. Sherman et al., 2017) were developed and widely used to investigate contaminant migration in porous media. Early studies were mainly based on assuming solute transport to be Fickian in porous media and fracture networks. Thus, the advection-dispersion equation was extensively applied to model transport in both homogeneous and heterogeneous soils and rocks. However, later investigations demonstrated that even in homogenous media, solute transport may conform to non-Fickian behavior (Aronofsky & Heller, 1957; Bromly & Hinz, 2004; Levy & Berkowitz, 2003; Scheidegger, 1959; Silliman & Simpson, 1987).

Evidence indicates that the ADE is not a satisfactory model for solute transport in inhomogeneous porous media (Berkowitz & Scher, 1995; Cortis & Berkowitz, 2004; Hunt et al., 2011). It predicts a Gaussian distribution for solute arrival times. However, both experimental and numerical results typically follow nearly power-law tails at large time scales, which cannot be explained by the ADE. Several alternative theories were proposed to model transport; among them, the continuous time random walk (CTRW) was widely applied. Although the generality of the CTRW makes it an attractive and promising choice for modelers, values for the specific parameters governing the truncated power-law arrival time distribution are unknown, which hinders its use for prediction purposes.

Solute transport in two- and three-dimensional fracture networks have been widely investigated experimentally and numerically. In addition to that, the effect of scale on dispersion has been well studied (Gao et al., 2010; Ghanbarian-Alavijeh et al., 2012; Pickens et al., 1981; Rajaram & Gelhar, 1993). However, the literature still lacks in a comprehensive analysis addressing the dependence of solute transport parameters and more specifically the continuous-time random walk (CTRW) model parameters on the scale, particularly for the sake of predictions. Therefore, the main objectives of this study are to: (1) generate three-dimensional fracture networks of various sizes, 20, 35, and 50 m representing tight geologic formations in nature considering the density of fracture p30 (total number of fractures per unit volume) in the systems, (2) investigate effect of domain size (scale) on fluid flow and solute transport, and (3) link solute transport parameters, e.g., average particle velocity (v), particle dispersion coefficient (D), characteristic transition time (t_1), cut-off time (t_2), and exponent β in the CTRW-TPL framework to fracture network topological and structural properties, e.g., domain size (L), density (p30 = N_f/L in which Nf is the total number of fractures in the network and L is the domain size), porosity (p33), surface area (p32), and fracture length-aperture distributions controlled by exponent α, and (4) finally examine the correlation between truncated power law distribution exponent α, effective permeability of the network and solute transport.

Chapter 2 - Numerical Simulations

In this study, first we stochastically generated discrete fracture networks. Then, we meshed them and solved the governing equations of flow (i.e., Stokes) to determine the velocity field. After that, we applied the Lagrangian approach of particle tracking to simulate solute transport and more specifically the arrival time distributions. Further details are provided in the following.

2.1. Fracture network generation

To generate the fracture networks, we used the data of Vermilye and Scholz (1995) who reported fracture aperture and length values for a dark-reddish siltstone outcrop in Culpeper Crushed Stone Quarry in Stevenburgs, Virginia. Overall, 23 fractures with lengths ranging from 65 mm to 13.2 m were collected. The local aperture value was measured every 5 mm along each fracture, and its average value ranged from 0.6 mm to 5 mm. Vermilye and Scholz (1995) found a positive correlation between fracture length and aperture. We accordingly assumed that fracture length and aperture were correlated in this study.

Three-dimensional fracture networks were then stochastically generated as shown in Fig. 2.1 at two fracture densities, defined as the sum of fracture lengths per unit volume, p30 = 0.05 and 0.1 m/m^3. Within the dfnWorks framework, each fracture is characterized by its length, orientation, aperture, and position of the fracture center. The fracture networks consisted of a single family whose orientation followed a von Mises-Fisher distribution and sampled using the algorithm provided by Wood (1994)

$$f(x;\mu,\kappa) = \frac{\kappa exp(\kappa \mu^T x)}{4\pi sinh(\kappa)} \quad (1)$$

where μ is the vector that defining the mean orientation, which can be expressed in terms of spherical coordinates (θ and ϕ), T denotes the transpose, $\kappa \geq 0$ is the concentration parameter that determines the degree of clustering around the mean direction ($\kappa \to 0$ results in a uniform distribution of fractures). Fractures in the networks were represented as planar polygon (ellipse) whose radii r followed a truncated power-law distribution with exponent α and lower and upper bounds $(r_0; r_u)$

$$Pr(r, r_0, r_u) = \frac{\alpha}{r_0} \frac{\left(\frac{r}{r_0}\right)^{-1-\alpha}}{1 - \left(\frac{r_u}{r_0}\right)^{-\alpha}} \quad (2)$$

We set r_0 = 1.5m and r_u = 6.75m, following Vermilye and Scholz (1995), and α = 1.5, 2.0, and 2.5, following Bonnet et al. (2001). A minimum feature size of *h* = 0.15 m (the one-tenth of the minimum fracture radius) was used in this study ensured any smaller fractures less than the value of *h* be removed during the network generation process.

Isolated clusters and fractures were removed after the generated networks, which resulted in slightly different p30 values. To address the effect of scale on flow and transport, cubic domains of the linear size L = 20, 35, and 50 m were investigated. Overall, we generated 18 fracture networks (3 α × 2 $p30$ × 3 L). Table 2.1 summarizes the parameters used to generate fracture network in this study.

Table 2.1. Parameters and their values used in this study to generate eighteen fracture networks. Fracture length and aperture data are from Vermilye and Scholz (1995)**.**

L (m)	$p30$	α	r_0(m)	r_u(m)	b_0(m)	b_u(m)	κ	ϕ	θ
20	0.05	1.5	1.5	6.75	0.0003	0.0025	1	0	0
35	0.10	2.0	-	-	-	-	-	-	-
50	-	2.5	-	-	-	-	-	-	-

L: domain size, p30: fracture density, α: truncated power-law exponent for fracture length distribution, r_0: minimum fracture radii, r_u: maximum fracture radii, b_0: minimum aperture, b_u: maximum aperture, κ :concentration parameter, ϕ: mean trend, θ: mean plunge

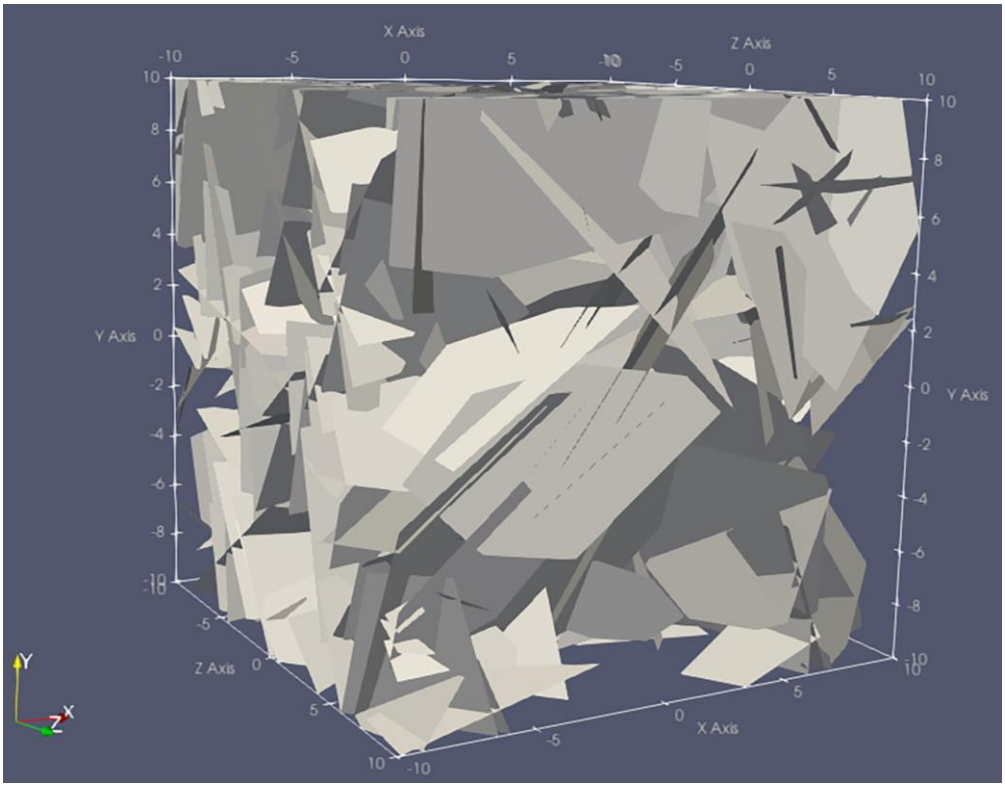


Figure 2.1. A DFN generation of L = 20 m with a total number of three hundred and thirty-eight fractures at p30 = 0.10 with $\boldsymbol{\alpha}$ = 1.5.

2.2. Network meshing

Each generated fracture network was meshed individually while preserving the polygon boundary line and segment corresponding to fracture/fracture intersections (see Fig. 2.2). The meshing was performed using the feature rejection algorithm for meshing (FRAM) and the (Los Alamos Grid toolbox (LaGrit) software package within the dfnWorks computational suite (J. D. Hyman et al., 2014, 2015). By linking network generation and meshing, FRAM creates an unstructured conforming Delaunay triangulation of each DFN without user intervention or manual mesh modification. At fracture intersections, the conforming Delaunay triangulation produces Voronoi control volumes suitable for finite volume solvers, e.g., finite element heat and

mass flow (FEHM), and parallel flow and reactive transport model (PFLOTRAN) (J. D. Hyman et al., 2014, 2015; Zyvoloski & Vesselinov, 2006).

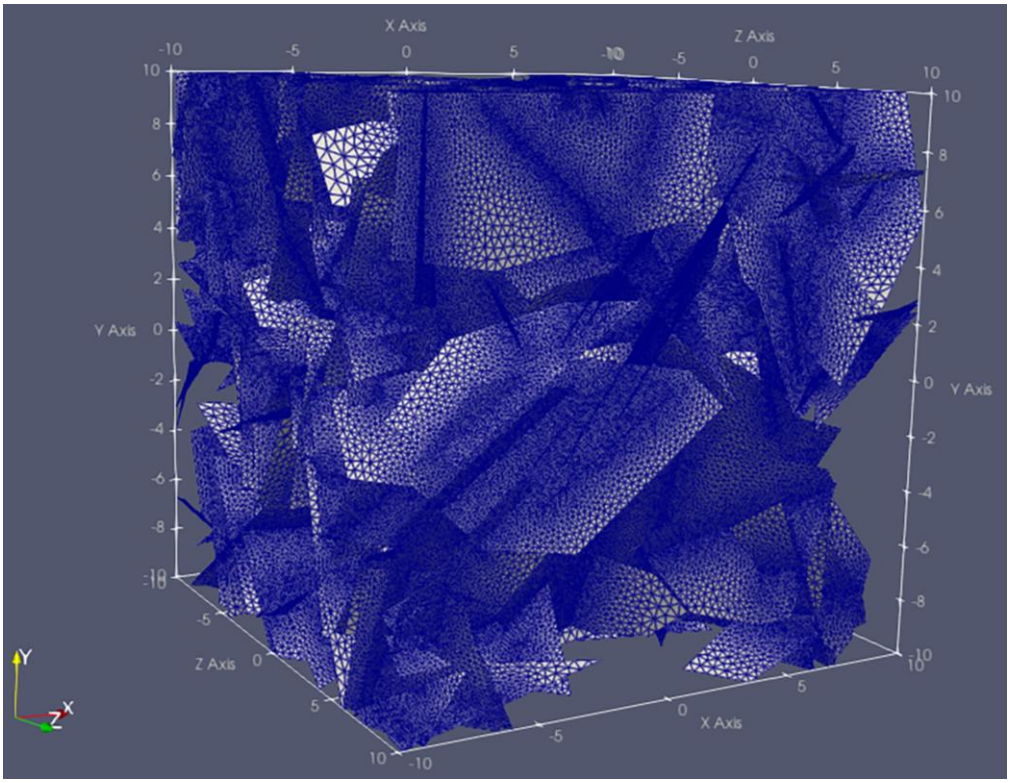


Figure 2.2. Meshing of DFN of L = 20 m at p30 = 0.10 with $\boldsymbol{\alpha}$ = 1.5. The mesh is coarsened away from the fracture intersection.

2.3. Fluid flow simulation

Fluid flow through the networks was simulated using the dfnFlow code and by computing the control volumes (Voronoi polygons), the dual mesh of the Delaunay triangulation after the network was generated and meshed. The dfnFlow code utilizes the massively parallel subsurface flow and reactive transport finite volume code PFLOTRAN, which can be used to integrate the governing flow equations numerically (J. D. Hyman et al., 2015; Jeffrey D. Hyman, 2020). We considered isothermal, incompressible, Newtonian fluid at the steady state through the fracture networks, which are governed by the Stokes equations. Assuming that the fracture

walls are parallel and smooth, the Stokes equations can be integrated to quantify the volumetric flow rate *Q* per unit fracture-aperture normal to the flow direction to obtain the Boussinesq equation. PFLOTRAN code solves the Reynolds partial differential equation for the pressure field in the network using the pressure conditions on the fracture domain boundaries, and this was then used to obtain a steady-state pressure solution to the fully saturated flow in the networks. (J. D. Hyman et al., 2015; Peter K. Kang et al., 2020; Zimmerman & Bodvarsson, 1996). Dirichlet boundary conditions were imposed at the inlet (2MPa) and outlet (1Mpa) as displayed in Fig. 2.3 below to create a pressure gradient along the z-direction, and no-flow boundary conditions were set along other boundaries. Local mass balance was maintained within the networks by utilizing a control volume-based discretization for flow, and flux evaluation accuracy was maintained using a Voronoi mesh (J. D. Hyman et al., 2015; Zyvoloski & Vesselinov, 2006). In this study, matrix diffusion is not considered in our simulations because the matrix surrounding the fractures is believed to be impervious, and there is no coupling between flow within fractures and the solid matrix.

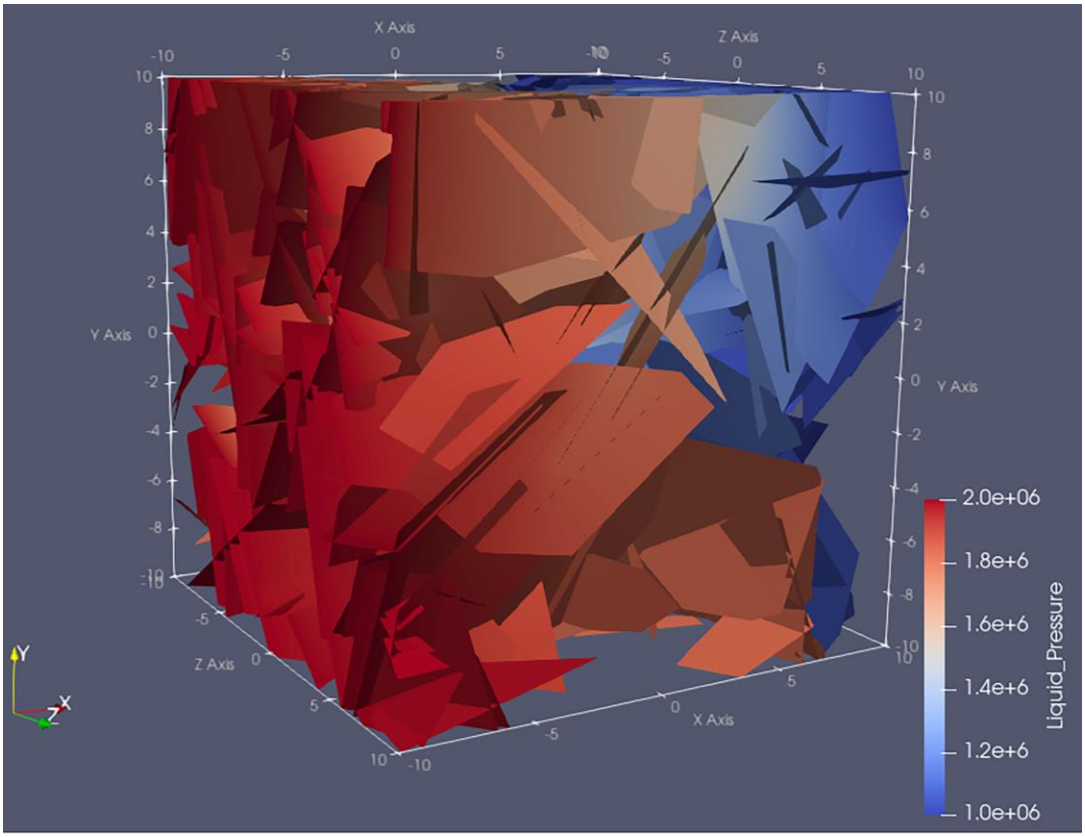


Figure 2.3. Steady- state pressure solution inside DFN of L = 20 m at p30 = 0.10 with $\boldsymbol{\alpha}$ = 1.5. The pressure boundary at the right and left plane of the network are 2 MPa and 1 MPa respectively.

2.4. Particle tracking simulation

We simulated mass transport by the dfnTrans code using control volume flow solutions and the unstructured mesh generated by dfnGen. The Darcy flux (including pressure and volumetric flow rate) from dfnFlow were used to determine the Eulerian velocity field. For details on particle tracking, see Painter et al. (2012) and Makedonska et al. (2015). Complete mixing of contaminants was assumed to occur at the fracture intersections instead of streamline routing. After determining the fluxes, we simulated particle transport through the network by the Lagrangian approach using a non-reactive conservative solute as a collection of indivisible passive tracer particles that trace the pathline through the velocity field. Particles were released

over a plane at the inlet plane perpendicular to the net flow direction of the domain, which was the x-axis, and, thus, particles were injected at a point x = 0, and at time $t = 0$ using the flux weighted injection method.

Although complex dynamics at fracture intersections may significantly affect mixing-induced reactions, they are not yet explicitly resolved (Jeffrey D. Hyman, 2020; Lee & Kang, 2020; Thomas Sherman et al., 2019). Under this condition, imposing a complete mixing model, which has negligible effects on longitudinal spreading, is the appropriate representation, and the probability of particles exiting into a fracture is weighted by the outgoing flux (Peter K. Kang et al., 2015, 2020; Park et al., 2001; Thomas Sherman et al., 2019).

For each realization, 50,000 particles were injected to each network and tracked from the inflow to the outflow plane boundary of the domain. Flow and transport simulations were iterated at least twenty times for each fracture network. Different realizations resulted in different arrival time distributions. To determine the appropriate number of realizations, we first iterated the simulations ten times and determined the slope of the arrival time distributions by fitting a power law at larger time scales. We thereafter calculated the mean and standard deviation of the slopes. We next increased the number of iterations to twenty and followed the same procedure to calculate the slopes, means and standard deviations for all twenty iterations. If no significant difference was observed between the results of ten and twenty iterations, we stopped. Otherwise, we increased the number of simulations to thirty and beyond until the results were insignificantly different. After that, to detect outliers, we computed the 95% confidence interval of the average arrival time distribution and removed those fell outside the 95% confidence interval boundaries.

Chapter 3 - Continuous time random walk model

Continuous-time random walk (CTRW) is one of the promising theories developed to study Fickian and non-Fickian transport. It is a generalization of the random walk method proposed first to describe diffusion. In fracture networks, contaminants migrate through the tortuous path of interconnected fractures. The motion can be envisioned as solutes executing a series of discrete transitions between fracture intersections. The size distribution of the fracture fragments sets the spatial scale; however, the transport is primarily controlled by the distribution of transit times in fracture fragments due to flow velocities. Anomalous transport behavior occurs if the velocity distribution is broad, i.e., there is a slow algebraic power tail for long arrival times. A sufficient interplay between the migrating solute along a flow path and a broad spectrum can produce this tail. The CTRW framework accounts naturally for the cumulative effects of a sequence of these transitions characterized by the following truncated power-law (TPL) distribution of the local transition times

$$\psi(t) = \left[t_1 \tau_2^{-\beta} \exp({\tau_2}^{-1}) \Gamma(-\beta, \tau_2^{-1})\right]^{-1} \frac{\exp(-t/t_2)}{(1 + t/t_1)^{1+\beta}} \propto \exp(-t/t_2) \qquad (3)$$

where t_1 and t_2 are respectively the lower and upper cutoffs of the truncated distribution (t_2 also represents the onset for the transition to Fickian transport), $\tau_2 \equiv t_1/t_2$, and $\Gamma(a, x)$ is the incomplete Gamma function. In Eq. (3), $\psi(t)$ follows a power-law trend like $(t/t_1)^{-1-\beta}$ for $t_1 \leq t \leq t_2$ i.e., in this time regime the transport behavior is anomalous, while $\psi(t)$ decreases exponentially for $t \geq t_2$, and the transport evolves from anomalous to normal transport. The exponent β controls the overall shape of breakthrough curve and is a useful means to characterize the cutoff t_2 range. In general, non-Fickian transport arises for $0 < \beta < 2$, for $0 < \beta < 1$, transport is highly non-Fickian and for $\beta > 2$, the transport becomes Fickian (Berkowitz

& Scher, 2009). Eq. (3) has been successfully used to account for many laboratory-based experiments, field observations, and behavior determined by simulations (Berkowitz, 2002; Cortis & Berkowitz, 2005).

In order to quantify the solute transport behavior in the generated fractured media, we applied the CTRW model. Five transport parameters $v, D, t_1, t_2,$ and β were optimized by fitting the CTRW model to the individual arrival time distributions (Fig. 4.2). For this purpose, we used the CTRW MATLAB toolbox developed by Cortis & Berkowitz, (2005). The initial guess for the average solute velocity was determined from the average linear velocity (Freeze & Cherry, 1979) and using the porosity and permeability data. As an initial guess for the dispersion coefficient and β, we used 10% of the average linear velocity and a value near 1 (due to heavy-tailed arrival time distributions), respectively. For t_1, we used a value two orders of magnitude smaller than the lower limit of the arrival time distribution, while for t_2 a value two orders of magnitude greater than the upper limit.

Chapter 4 - Results and Discussion

4.1. Effect of exponent α on effective permeability

To investigate the impact of the exponent α, we plotted the effective permeability values averaged over all the realizations for each network against α in Fig. 4.1. Results for two fracture densities p30 = 0.05 and 0.10 are also summarized in Table 4.1. As can be observed from Fig. 4.1, with increasing α the value of k_{eff} decreases. The exponent α controls the frequency of longer and shorter fractures in the network (de Dreuzy et al., 2012). As the value of exponent α increases, a larger number of small fractures dominates the network. Thus, the k_{eff} of the network decreases with increasing the α values. Zhu et al. (2021) observed similar results of increasing exponent α on flow rates in three-dimension DFN simulations. Since the apertures are positively correlated to the fracture lengths in our study, long fractures tend to contribute more to the fluid flow due to their large apertures. We found that the k_{eff} strongly depended on the number and permeability of individual fractures that cut across the entire network rather than the total number of fractures in the networks. Particularly in sparse networks with low densities studied here, a few long fractures may significantly contribute to the permeability in the system. For p30 = 0.05 and α = 1.5, we observed one order of magnitude increase in the k_{eff} value higher than α = 2.0 even though the number of fractures in the network with α = 2.0 was higher than that with α = 1.5. We also found a direct proportionality between the exponent α and the fracture surface area (also known as fracture intensity p32). Similar trend was reported by Hyman et al. (2019). The exponent α controls the distributions of fracture surface area, which does influence the permeability distributions, specifically in sparse networks considered in this study, more than the number of fractures in the systems (e.g., Table 4.1 for p30 = 0.10 and

exponent α = 1.5). Recall that we generated the fracture networks stochastically, and it has been reported that realistic fracture networks share most of the attributes of random fracture networks (Zhu et al., 2021). All the fracture networks in this study mimic that of real-world fracture networks in terms of geometry. Specifically, the fracture intensity p32 in this study, which represents the total fracture surface per unit volume ranges from 1.03 to 1.88 m^{-1} (Table 4.1). This p32 value is consistent with p32 measurement of core data sets from New Albany Shale Unit, Baxter Shale Unit, and Marcellus Shale Unit (Gale et al., 2017).

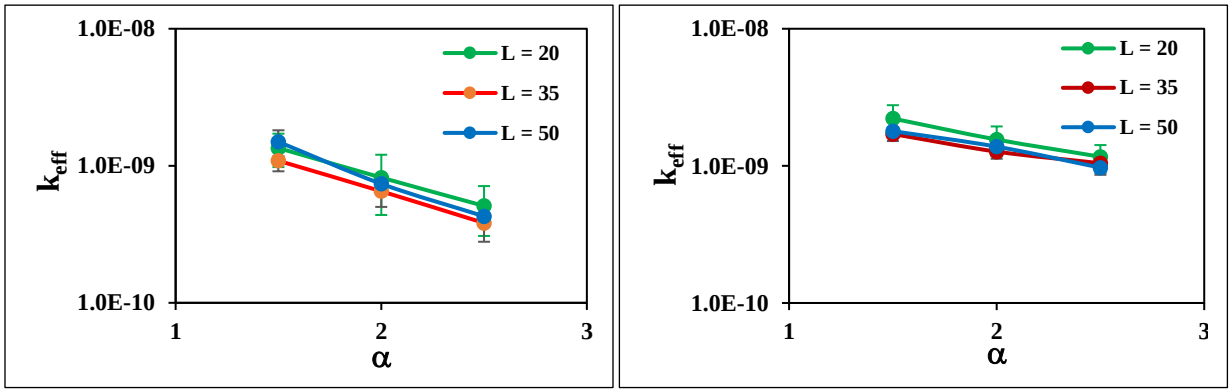


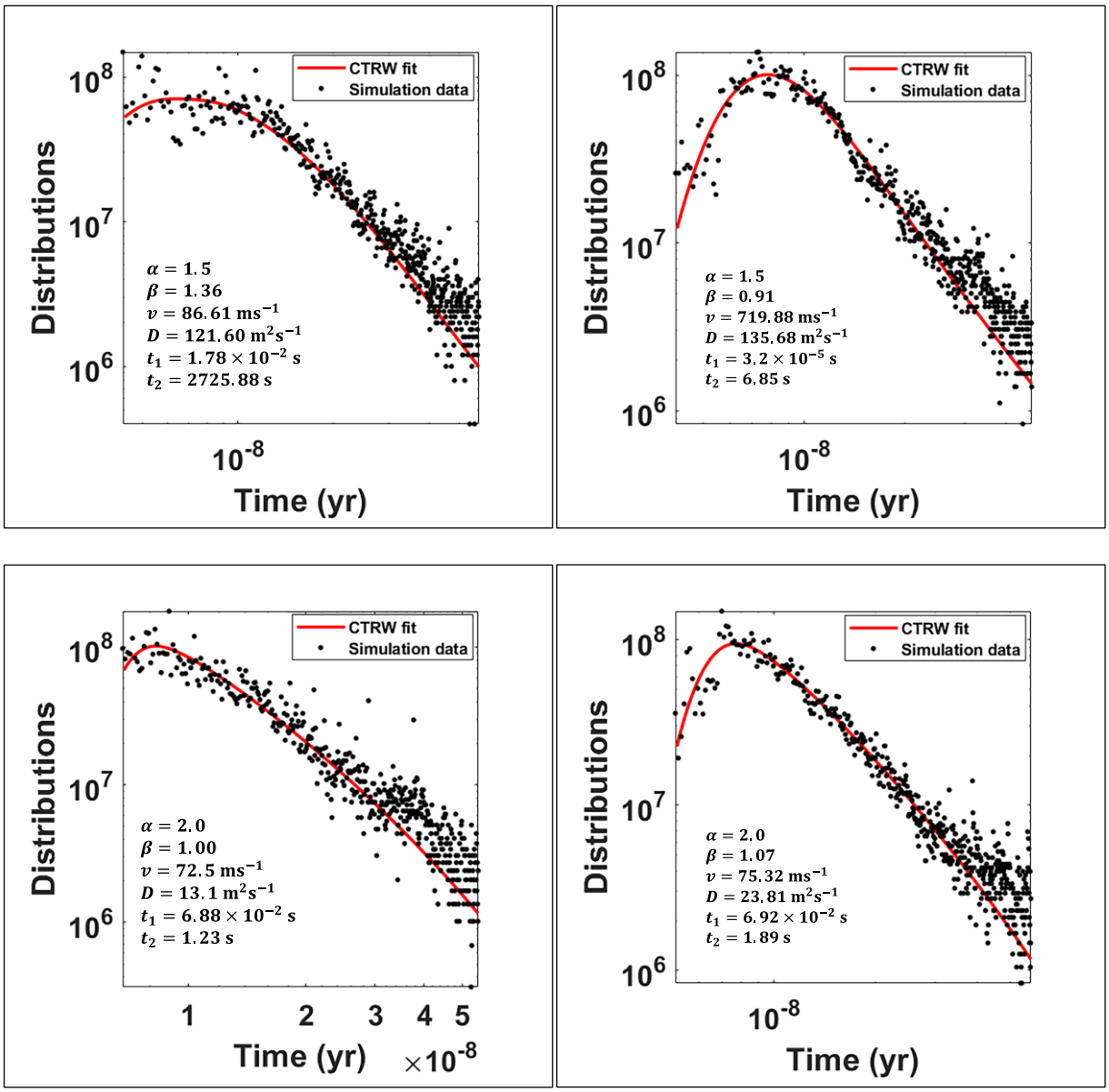
Figure 4.1. Permeability *k* as a function of exponent $\boldsymbol{\alpha}$ for (left) p30 = 0.05 and (right) p30 = 0.10 plotted for different domain sizes (*L* = 20, 30, and 50 m)

4.2. Effect of exponent α on solute transport

Although the effect of α on fluid flow and permeability in fracture networks has been previously investigated (Bogdanov et al., 2007; de Dreuzy et al., 2001), its influence on solute transport and more specifically non-Fickian behavior in tight media has not well addressed yet.

The average values of the optimized CTRW model parameters for 18 fracture networks studied here are summarized in Table 4.1. The standard deviation for each parameter is given in Table A1 in Appendix A. The CTRW fit to the simulated arrival time distribution is shown in Fig.4.2 for six fracture networks depicted in Fig. A.1 in Appendix A. As can be seen, the CTRW

reasonably fit the arrival time distributions. Similar results were obtained for other twelve sets of simulations (not shown). Overall, we found $0.69 \leq \beta \leq 1.25$ indicating non-Fickian solute behavior (Berkowitz et al., 2006). The long tails at large time scales shown in Fig. 4.2 confirm the non-Fickian transport behavior. Regarding the origin of non-Fickian transport in fracture networks, Neuman (2005) attributed that to flow channeling and fast paths through fractured media.



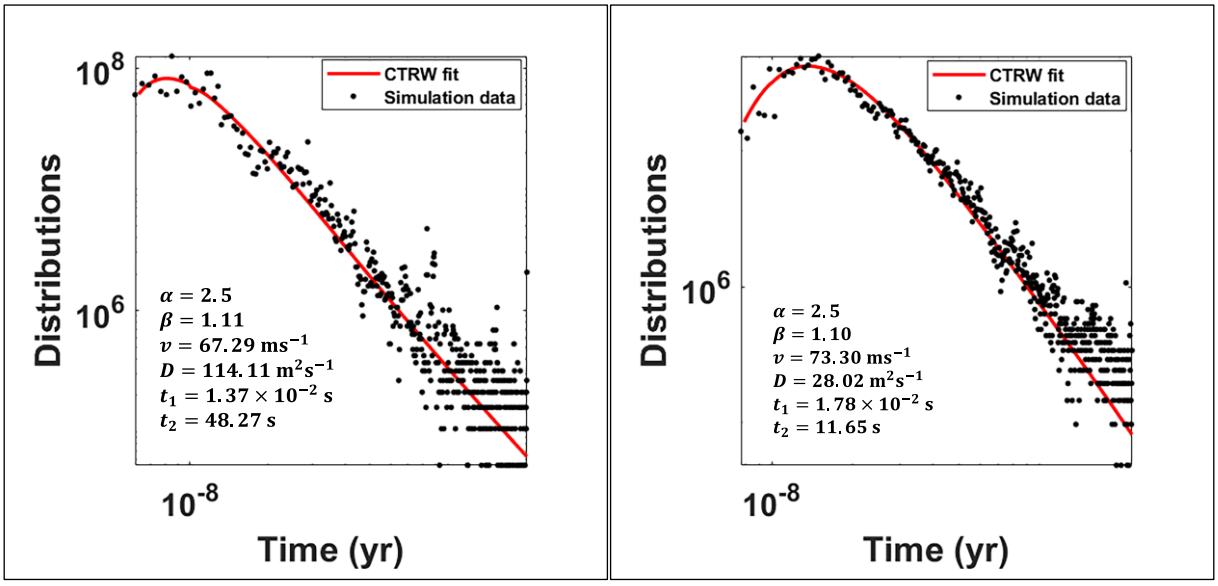
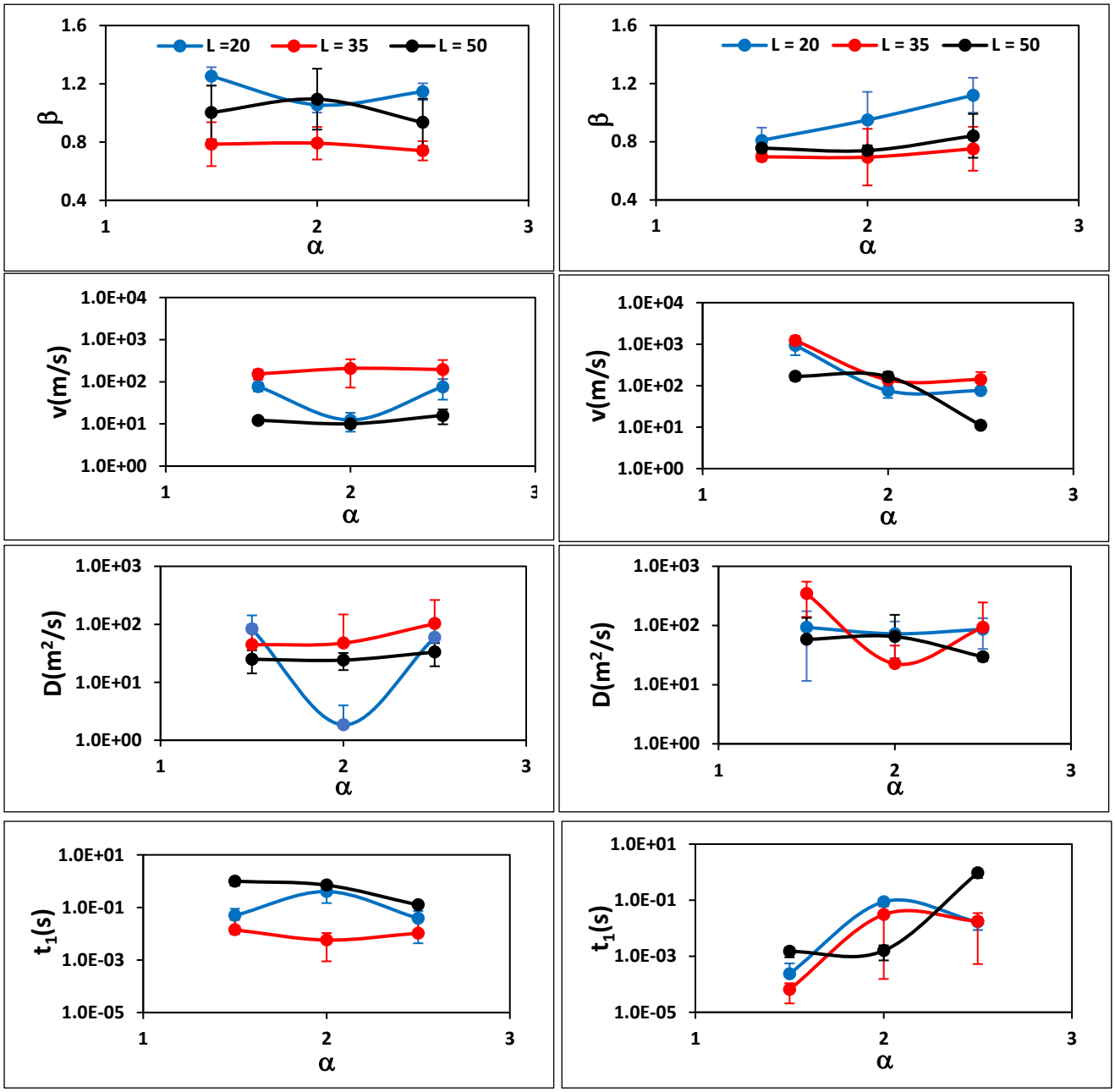


Figure 4.2. Simulated arrival time distributions and fitted CTRW model for a single DFN realization and L = 20 m. Left plots correspond to p30 = 0.05 and right ones to p30 = 0.10.

In Fig. 4.2.1, we show the average values of the CTRW model parameters β, v, D, t_1 and t_2 versus α for two fracture densities studied here (p30 = 0.05 and 0.1). We did not find any monotonic relationship between the average β values reported in Table 4.1 and α. For example, for p30 = 0.05 and L = 20 and 35 m the average value of β decreased monotonically with increasing α. However, for other cases β first increased as α increased from 1.5 to 2 and then decreased with increasing α from 2 to 2.5 (see Table 4.1). Similar results were observed for the other CTRW model parameters (Fig. 4.2.1).

We explored the relationship between β and α in further detail. By averaging over all six values corresponding to two fracture densities and three system sizes, we found β = 0.89, 0.89, and 0.92 for α = 1.5, 2, and 2.5, respectively, showing as α increased, on average the value of β increased as well. Although similar results were found within the networks corresponding to p30 = 0.1, the trend between β and α was decreasing in the networks with p30 = 0.05. Further

investigations and simulations at various fracture densities are still required to better address how varying α affects the value of β and solute transport in fracture networks.



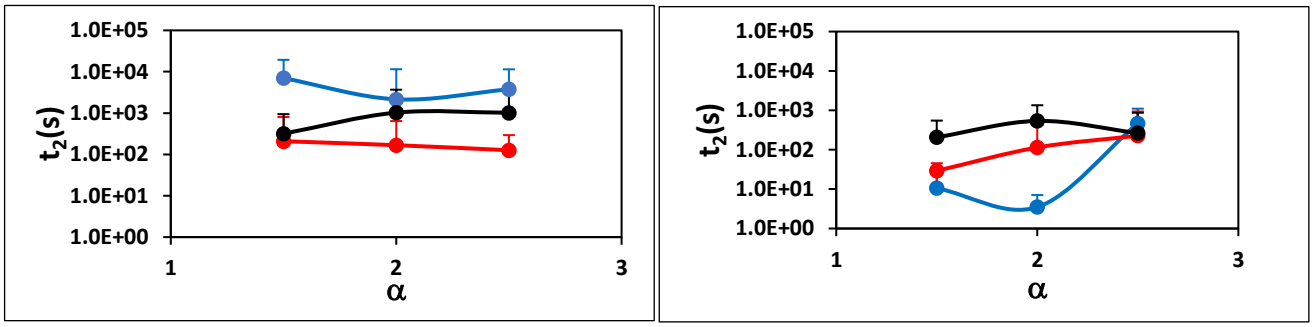


Figure 4.2.1. The CTRW model parameters against exponent $\boldsymbol{\alpha}$ for p30 = 0.05 (left) and 0.1 (right)

4.3. Effects of fracture density and scale on fluid flow

We found that in the fracture networks corresponding to the same α value, as the fracture density increased, the value of effective permeability increased as well (see Table 4.1). The increasing trend varied from one α value to another. On average, we found that the value of permeability increased from p30 = 0.05 to 0.1 by a factor of 1.5, 1.9 and 2.4 for α = 1.5, 2, and 2.5, respectively.

Results of the scale dependence of k_{eff} for two fracture densities 0.05 and 0.1 are shown in Fig. 4.3. For p30 = 0.05, we found that k_{eff} slightly decreased by 19, 26, and 25% respectively for α = 1.5, 2, and 2.5 as L increased from 20 to 35 m. Its value then slightly increased by 37, 13, and 12% for α = 1.5, 2, and 2.5, respectively, with increasing L from 35 to 50 m. For p30 = 0.1, the k_{eff} dropped slightly by 23% for α = 1.5, 18% for α = 2, and 10% for α = 2.5 when L increased from 20 to 35. When the system size increased from 35 to 50 m the value of k_{eff} increased by 5 and 9% for respectively α = 1.5 and 2, while decreased by 7% for α = 2.5. Overall, our results showed less scale-dependent k_{eff} for p30 = 0.1 compared to 0.05. This is most probably because at the fracture density of 0.05, the generated networks are closer

to the percolation threshold where a disordered sample spanning cluster forms, and one should expect a very high level of heterogeneity.

The fluid flow simulations and the k_{eff} values presented in Table 4.1 and depicted as a function of the system size in Fig. 4.3 showed that the generated networks with size 50 m for both p30 = 0.05 and 0.1 should be close to the representative elementary volume (REV).

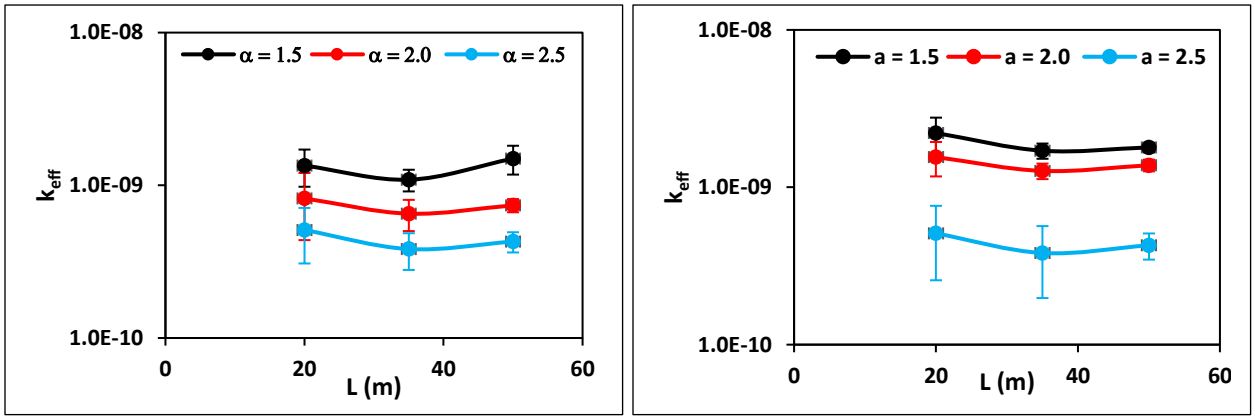


Figure 4.3. Effective permeability, k_{eff}, against network size, L, for p30 = 0.05 (left) and 0.1 (right) and three $\boldsymbol{\alpha}$ values. The data are summarized in Table 4.1.

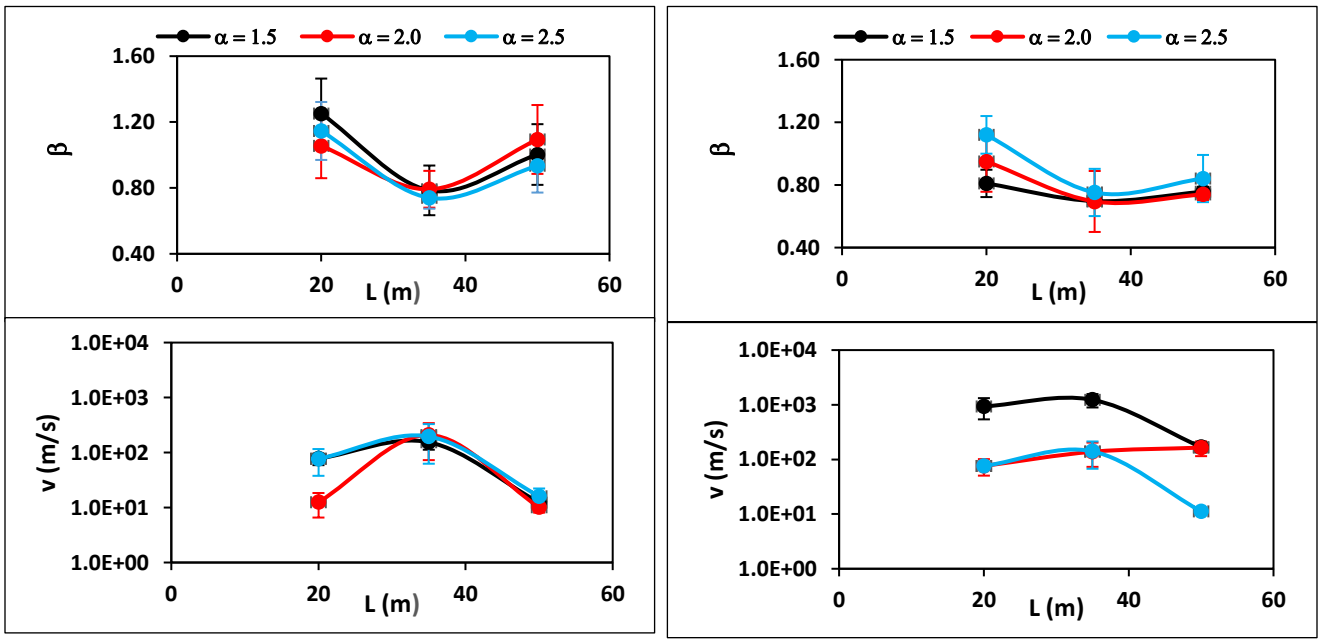
4.4. Effects of fracture density and scale on solute transport

In Fig. 4.4, we show the CTRW model parameters as a function of the system size L for p30 = 0.05 and 0.10 and different α values. As can be seen, all the CTRW model parameters showed strong scale-dependent behavior, although the effect of scale on the k_{eff} was relatively trivial (Fig. 4.3). Plots corresponding to p30 = 0.05 are as scattered as those corresponding to p30 = 0.10, which makes it difficult to determine at which fracture density (0.05 or 0.1) the effect of scale on solute transport is more substantial. Fracture networks with low fracture densities are very heterogeneous and complex systems, which makes solute transport simulations very challenging and computationally demanding (Cherubini et al., 2022; Davy et al., 2006).

Generally speaking, we found slightly smaller β values ($0.69 \leq \beta \leq 1.12$) in the higher density fracture networks (p30 = 0.1) than those ($0.74 \leq \beta \leq 1.25$) in the lower density ones (p30 = 0.05). Although one may expect β to be smaller in lower density networks, due to the presence of higher level of heterogeneity, the two ranges of β are not greatly different. For both p30 = 0.05 and 0.1, small values of β (< 1.25) indicate strong non-Fickian transport behavior in such networks and heavy tails in the arrival time distributions.

We also found that the average solute velocity (v) in the networks with p30 = 0.1 was generally greater than that in the networks with p30 = 0.05. These results are consistent with the results of fluid flow simulations for which we found the k_{eff} in the networks with higher fracture densities greater than that in the networks with lower fracture densities (Table 4.1).

Our results showed that by increasing v, the value of dispersion coefficient D also increased with $R^2 = 0.64$ (results not shown). Similar to the results of Ghanbarian et al. (2023), we found as either v or D increased, the value of t_1 decreased with $R^2 = 0.82$ (results not shown).



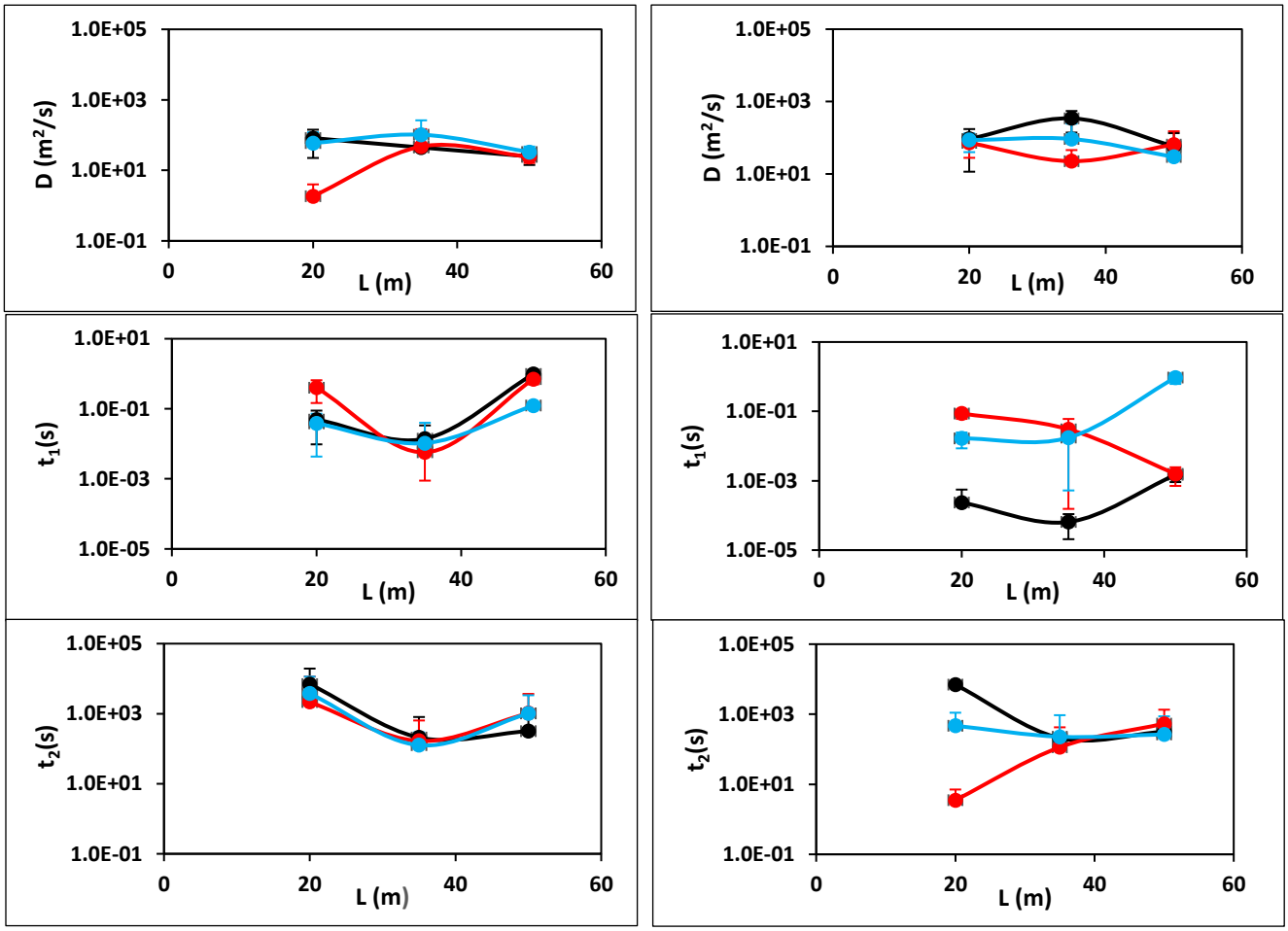


Figure 4.4. CTRW model parameters against domain size for (left) p30 = 0.05 and (right) p30 = 0.1. Error bars shows the variabilities from various realizations. The data in the tabulated format are summarized in Table 4.1.

4.5. Regression-based relationships for the CTRW model parameters

In this section, we investigated the correlation between the geometrical and topological properties of the fracture networks, e.g., density (p30), fracture surface area (p32), network porosity (p33), domain size (L), exponent (α), effective permeability (k_{eff}), and the optimized CTRW model parameters. We first examined the normality of the input and output variables using the Jarque-Bera test. If not normally distributed, we used the Cox-Box transformation and verified the normality of the transformed variables via the Jarque-Bera test. We then applied

multiple linear regression analysis in R to link the fracture network geometrical and topological properties to the CTRW model parameters. Results summarized in Table 4.5 indicate that $0.74 \leq R^2 \leq 0.84$ and $0.0002 \leq$ p-value ≤ 0.0085. No model was found for the dispersion coefficient with high R^2 or p-value < 0.05 (at 95% confidence level). Although through multiple linear regression analysis, we found $\beta, v,$ and t_1 to be significantly dependent on the system size L (p-value < 0.05), the parameter t_2 was only linked to p32, p33, N_f, and α (Table 4.5). The reason is still not clear, and further investigations are required.

Generally speaking, one should expect the average solute velocity to increase as the effective permeability increases. This is consistent with our results given in Table 4.1 and the regression-based model presented for v. In addition to that, earlier we showed that as α increased, on average the value of β increased as well. The regression model for β, reported in Table 4.5, similarly mimic the same trend between β and α. Our regression analysis also showed an inverse relationship between the v and the exponent α, which seems reasonable because solutes would transit at a faster rate when large fractures dominate and cross a network.

We should point out that the β showed more sensitivity to the number of fractures (N_f) in the networks than the p30. Recall that after fractures were generated, isolated fractures and clusters were removed from the networks because they did not contribute to flow and transport led to a different number of fractures at each realization. Based on our regressions analysis, as the number of fractures increases, the value of β also increases, i.e., the networks tend to be homogeneous with increasing number of fractures.

Table 4.1. Fracture geometrical and topological properties from fracture network generations, and averaged CTRW model optimized parameters obtained from the fits of the model to arrival time distributions of particle tracking simulations.

Fracture network topological properties							**CTRW model parameters**				
P30 ($\mathbf{m^{-2}}$)	L**(m)**	$\mathbf{N_f}$	**P33** $\times$ $\mathbf{10^{-3}}$	**P32** ($\mathbf{m^{-1}}$)	α	$K(\mathbf{m^2})$	β	$v(\mathbf{ms^{-1}})$	$D(\mathbf{m^2s^{-1}})$	$t_1(\mathbf{s})$	$t_2(\mathbf{s})$
0.05	20	193	1.61	1.30	1.5	1.35×10^{-09}	1.25	7.74×10^{01}	8.25×10^{01}	4.92×10^{-02}	6.99×10^{03}
		184	1.34	1.20	2.0	8.21×10^{-10}	1.06	1.25×10^{01}	1.85×10^{00}	4.00×10^{-01}	2.15×10^{03}
		166	1.17	1.03	2.5	5.09×10^{-10}	1.15	7.67×10^{01}	5.95×10^{01}	3.84×10^{-02}	3.76×10^{03}
	35	1011	1.51	1.33	1.5	1.09×10^{-09}	0.79	1.54×10^{02}	4.45×10^{01}	1.04×10^{-02}	2.09×10^{02}
		993	1.29	1.28	2.0	6.52×10^{-10}	0.79	2.08×10^{02}	4.73×10^{01}	5.74×10^{-03}	1.67×10^{02}
		974	1.06	1.21	2.5	3.82×10^{-10}	0.74	1.96×10^{02}	1.03×10^{02}	1.05×10^{-02}	1.26×10^{02}
	50	3124	1.55	1.44	1.5	1.50×10^{-09}	1.00	1.23×10^{01}	2.49×10^{01}	9.88×10^{-01}	3.17×10^{02}
		3153	1.39	1.40	2.0	7.39×10^{-10}	1.09	1.01×10^{01}	2.41×10^{01}	7.04×10^{-01}	1.02×10^{03}
		3139	1.19	1.34	2.5	4.28×10^{-10}	0.94	1.60×10^{01}	3.31×10^{01}	1.24×10^{-01}	1.02×10^{03}

0.10	20	338	2.08	1.77	1.5	2.21×10^{-09}	0.81	9.31×10^{02}	9.30×10^{01}	2.35×10^{-04}	1.05×10^{01}
		342	1.77	1.74	2.0	1.56×10^{-09}	0.95	7.58×10^{01}	7.24×10^{01}	8.69×10^{-02}	3.52×10^{00}
		338	1.79	1.65	2.5	1.17×10^{-09}	1.12	7.58×10^{01}	8.63×10^{01}	1.66×10^{-02}	4.68×10^{01}
	35	1648	1.97	1.75	1.5	1.71×10^{-09}	0.70	1.23×10^{03}	3.45×10^{02}	6.49×10^{-05}	2.92×10^{01}
		1679	1.71	1.70	2.0	1.27×10^{-09}	0.69	1.38×10^{02}	2.26×10^{01}	3.03×10^{-02}	1.14×10^{02}
		1676	1.59	1.69	2.5	1.05×10^{-09}	0.75	1.40×10^{02}	9.27×10^{01}	1.76×10^{-02}	2.28×10^{02}
	50	4972	1.98	1.88	1.5	1.79×10^{-09}	0.76	1.68×10^{02}	5.81×10^{01}	1.48×10^{-03}	2.07×10^{02}
		5052	1.85	1.86	2.0	1.38×10^{-09}	0.74	1.64×10^{02}	6.49×10^{01}	1.58×10^{-03}	5.39×10^{02}
		5107	1.67	1.82	2.5	9.71×10^{-10}	0.84	1.12×10^{01}	2.95×10^{01}	9.44×10^{-01}	2.65×10^{02}

p30: fracture density, L: domain size, N_f: final number of fractures after isolated fractures have been removed from network, p32: initial fracture surface area, α: truncated power-law exponent for fracture length distribution, K: system permeability; β: CTRW-TPL exponent, v: solute velocity, D: dispersion, t_1: transition time, t_2: cut-off time.

Figure 4.5. Basic regression model for CTRW model parameters

Variable	Equation	R^2	P-value
β	$(9.73 + 0.03L - 2.13\alpha - 4368.36p33 - 2.61p32 + 78.35p30)^{-1/2.36}$	0.62	0.0036
v	$\left(2.30 - 2.63 \times 10^{-5}N_f - 0.56p32 - 717.6p33 - 0.11\alpha - 1.32 \times 10^{8}k_{eff} + 507.30p32 \times p33\right)^{1/0.03}$	0.69	0.0048
t_1	$\left(-1.71 + 1.60 \times 10^{-4}N_f + 2.52p32 + 1396.20p33 - 1382.40p32 \times p33 - 0.01L \times p32 + 1.84 \times 10^{8}k_{eff}\right)^{1/0.06}$	0.67	0.0114
t_2	$\left(0.64 + 1.90 \times 10^{-4}N_f - 1081.20p32 \times p33 + 2002.20p33 - 0.01L + 5.16p30\right)^{1/0.06}$	0.78	0.0002

p30: fracture density (m^{-2}), p32: fracture surface area (m^{-1}), L: domain size (m), $k_{eff}(m^2)$, p33: network porosity, N_f: number of fractures in the network

Note: No high correlation was found for dispersion (D); thus, it was not reported in the table.

Chapter 5 - Conclusions

We investigated the effect of scale and truncated power-law fracture length distribution on both fluid flow and solute transport in three-dimensional fracture networks at two different fracture densities (p30 = 0.05 and 0.1) representing tight formations. We considered three values i.e., 1.5, 2, and 2.5 (consistent with experiments reported in the literature) for the exponent α, numerically solved the Stokes equation to determine the velocity field, and applied a Lagrangian approach to simulate advective transport through eighteen fracture networks. To quantify the solute transport behavior, we fit the CTRW model to the simulated arrival time distributions obtained from particle tracking simulations. Results showed that the generated networks were near the representative elementary volume (REV) for fluid flow. More specifically, the effect of scale (fracture network linear size L) on the effective permeability was trivial, particularly for L = 35 and 50 m. However, by investigating the scale dependence of the CTRW model parameters, we showed that the REV for solute transport has not reached. To the best of our knowledge, this is the first study demonstrating the REV for fluid flow is different from that for solute transport in tight formations. We also established scale-dependent regression-based relationships linking the CTRW model parameters to the system size and other fracture network characteristics. We found parameters α, p32, p33, N_f, and L significantly control the CTRW model parameters, and more generally the solute transport behavior (p-value < 0.05).

References

Aronofsky, J. S., & Heller, J. P. (1957). Aronofsky, J. S., and J. P. Heller (1957), A diffusion model to explain mixing of flowing miscible fluids in porous media, Trans. Am. Inst. Min. Metall. Pet. Eng., 210, 345–349. *Trans. Am. Inst. Min. Metall. Pet. Eng.*, *210*, 345–349.

Benson, D. A., Wheatcraft, S. W., & Meerschaert, M. M. (2000). Application of a fractional advection-dispersion equation. *Water Resources Research*, *36*(6), 1403–1412. https://doi.org/10.1029/2000WR900031

Berkowitz, B. (2002). Characterizing flow and transport in fractured geological media: A review. *Advances in Water Resources*, *25*(8–12), 861–884. https://doi.org/10.1016/S0309-1708(02)00042-8

Berkowitz, B., & Scher, H. (1995). On Characterization of Anomalous Dispersion in Porous and Fractured Media. *Water Resources Research*. https://doi.org/10.1029/95WR00483

Berkowitz, B., & Scher, H. (2009). Exploring the nature of non-Fickian transport in laboratory experiments. *Advances in Water Resources*, *32*(5), 750–755. https://doi.org/10.1016/j.advwatres.2008.05.004

Berkowitz, B., Cortis, A., Dentz, M., & Scher, H. (2006). Modeling non-Fickian transport in geological formations as a continuous time random walk. *Reviews of Geophysics*, *44*, RG2003.

Berkowitz, B., Dror, I., Hansen, S. K., & Scher, H. (2016). Measurements and models of reactive transport in geological media. *Reviews of Geophysics*, 930–986. https://doi.org/10.1002/2016RG000524.Received

Bogdanov, I. I., Mourzenko, V. V., Thovert, J. F., & Adler, P. M. (2007). Effective permeability of fractured porous media with power-law distribution of fracture sizes. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *76*(3), 1–15. https://doi.org/10.1103/PhysRevE.76.036309

Bonnet, E., Bour, O., Odling, N. E., Davy, P., Main, I., Cowie, P., & Berkowitz, B. (2001). Scaling of fracture systems in geological media. *Reviews of Geophysics*, *39*(3), 347–383. https://doi.org/10.1029/1999RG000074

Bromly, M., & Hinz, C. (2004). Non-Fickian transport in homogeneous unsaturated repacked sand. *Water Resources Research*, *40*(7), 5–9. https://doi.org/10.1029/2003WR002579

Cacas, M. C., Ledoux, E., de Marsily, G., Tillie, B., Barbreau, A., Durand, E., et al. (1990). Modeling fracture flow with a stochastic discrete fracture network: calibration and validation: 1. The flow model. *Water Resources Research*, *26*(3), 479–489. https://doi.org/10.1029/WR026i003p00479

Chen, Y. F., Fang, S., Wu, D. S., & Hu, R. (2017). Visualizing and quantifying the crossover from capillary fingering to viscous fingering in a rough fracture. *Water Resources Research*, *53*, 7756–7772.

Cherubini, C., Giasi, C. I., Pastore, N., Wang, L., Bayani Cardenas, M., Jiménez-Hornero, F. J., et al. (2022). Fractal Characterization of Multimodal, Multiscale Images of Shale Rock Fracture Networks. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*, *44*(4), 2165–2216. https://doi.org/10.1111/j.1745-6584.2006.00236.x

Coats, K., & Smith, B. (1964). Dead-end pore volume and dispersion in porous media. *SPE Journal*, *4*, 73–84. https://doi.org/10.2118/647-PA

Cortis, A., & Berkowitz, B. (2004). Anomalous Transport in "Classical" Soil and Sand Columns. *Soil Science Society of America Journal*, *68*(5), 1539–1548. https://doi.org/10.2136/sssaj2004.1539

Cortis, A., & Berkowitz, B. (2005). Computing "anomalous" contaminant transport in porous media: The CTRW MATLAB toolbox. *Ground Water*, *43*(6), 947–950. https://doi.org/10.1111/j.1745-6584.2005.00045.x

Davy, P., Bour, O., De Dreuzy, J. R., & Darcel, C. (2006). Flow in multiscale fractal fracture networks. *Geological Society Special Publication*, *261*, 31–45. https://doi.org/10.1144/GSL.SP.2006.261.01.03

Detwiler, R. L., Rajaram, H., & Glass, R. J. (2000). Solute transport in variable-aperture fractures: An investigation of the relative importance of Taylor dispersion and macrodispersion. *Water Resources Research*, *36*(7), 1611. https://doi.org/10.1029/2000WR900036

de Dreuzy, J. R., Davy, P., & Bour, O. (2001). Hydraulic properties of two-dimensional random fracture networks following a power law length distribution 1. Effective connectivity. *Water Resources Research*, *37*(8), 2065–2078. https://doi.org/10.1029/2001WR900011

De Dreuzy, J. R., Méheust, Y., & Pichot, G. (2012). Influence of fracture scale heterogeneity on the flow properties of three-dimensional discrete fracture networks (DFN). *Journal of Geophysical Research B: Solid Earth*, *117*(11), 1–21. https://doi.org/10.1029/2012JB009461

Faybishenko, B., Benson, S., & Gale, J. (2015). *Dynamics of Fluids and Transport in Complex Fractured-Porous Systems, Geophysical Monograph Series 210*. American Geophysical Union and Wiley.

Forstner, S. R., & Laubach, S. E. (2022). Scale-dependent fracture networks. *Journal of Structural Geology*, *165*, 104748.

Frampton, A., & Cvetkovic, V. (2011). Numerical and analytical modeling of advective travel times in realistic three-dimensional fracture networks. *Water Resources Research*, *47*(2). https://doi.org/10.1029/2010WR009290

Freeze, R. A., & Cherry, J. A. (1979). *Groundwater*. *Groundwater*. Retrieved from https://www.ptonline.com/articles/how-to-get-better-mfi-results

Gale, J. F. W., Laubach, S. E., Olson, J. E., Eichhubl, P., & Fall, A. (2017). Natural fractures in shale: A review and new observations. *AAPG Bulletin*, *101*(8), 2165–2216. https://doi.org/10.1306/08121413151

Gao, G., Zhan, H., Feng, S., Fu, B., Ma, Y., & Huang, G. (2010). A new mobile-immobile model for reactive solute transport with scale-dependent dispersion. *Water Resources Research*, *46*(8), 1–16. https://doi.org/10.1029/2009WR008707

van Genuchten, M. T., & Wagenet, R. J. (1989). Two-Site/Two-Region Models for Pesticide Transport and Degradation: Theoretical Development and Analytical Solutions. *Soil Science Society of America Journal*, *53*(October), 1303–1310. https://doi.org/10.2136/sssaj1989.03615995005300050001x

Ghanbarian-Alavijeh, B., Skinner, T. E., & Hunt, A. G. (2012). Saturation dependence of dispersion in porous media. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *86*(6), 1–14. https://doi.org/10.1103/PhysRevE.86.066316

Gong, J., & Rossen, W. R. (2017). Modeling flow in naturally fractured reservoirs: effect of fracture aperture distribution on dominant sub-network for flow. *Petroleum Science*, *14*, 138–154.

Grisak, G. E., & Pickens, J. F. (1980). Solute transport through fractured media: 1. The effect of matrix diffusion. *Water Resources Research*, *16*(4), 719–730. https://doi.org/10.1029/WR016i004p00719

Hull, L. C., Miller, J. D., & Clemo, T. M. (1987). Laboratory and simulation studies of solute transport in fracture networks. *Water Resources Research*, *23*(8), 1505–1513. https://doi.org/10.1029/WR023i008p01505

Hunt, A. G., & Ghanbarian, B. (2016). Percolation theory for solute transport in porous media: Geochemistry, geomorphology, and carbon cycling. *Water Resources Research*, *52*(9). https://doi.org/10.1002/2016WR019289

Hunt, A. G., Skinner, T. E., Ewing, R. P., & Ghanbarian-Alavijeh, B. (2011). Dispersion of solutes in porous media. *European Physical Journal B*, *80*(4), 411–432. https://doi.org/10.1140/epjb/e2011-10805-y

Hyman, J. D., Gable, C. W., Painter, S. L., & Makedonska, N. (2014). Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy. *SIAM Journal on Scientific Computing*, *36*(4), A1871–A1894. https://doi.org/10.1137/130942541

Hyman, J. D., Karra, S., Makedonska, N., Gable, C. W., Painter, S. L., & Viswanathan, H. S. (2015). DfnWorks: A discrete fracture network framework for modeling subsurface flow

and transport. *Computers and Geosciences*, *84*, 10–19. https://doi.org/10.1016/j.cageo.2015.08.001

Hyman, J. D., Dentz, M., Hagberg, A., & Kang, P. K. (2019). Linking Structural and Transport Properties in Three-Dimensional Fracture Networks. *Journal of Geophysical Research: Solid Earth*, *124*(2), 1185–1204. https://doi.org/10.1029/2018JB016553

Hyman, Jeffrey D. (2020). Flow Channeling in Fracture Networks: Characterizing the Effect of Density on Preferential Flow Path Formation. *Water Resources Research*, *56*(9), 1–21. https://doi.org/10.1029/2020WR027986

Kang, P. K., Dentz, M., Le Borgne, T., & Juanes, R. (2011). Spatial Markov model of anomalous transport through random lattice networks. *Physical Review Letters*, *107*, 180602.

Kang, Peter K., Dentz, M., Le Borgne, T., & Juanes, R. (2015). Anomalous transport on regular fracture networks: Impact of conductivity heterogeneity and mixing at fracture intersections. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *92*(2). https://doi.org/10.1103/PhysRevE.92.022148

Kang, Peter K., Dentz, M., Le Borgne, T., Lee, S., & Juanes, R. (2017). Anomalous transport in disordered fracture networks: Spatial Markov model for dispersion with variable injection modes. *Advances in Water Resources*, *106*, 80–94. https://doi.org/10.1016/j.advwatres.2017.03.024

Kang, Peter K., Lei, Q., Dentz, M., & Juanes, R. (2019). Stress-Induced Anomalous Transport in Natural Fracture Networks. *Water Resources Research*, *55*(5), 4163–4185. https://doi.org/10.1029/2019WR024944

Kang, Peter K., Hyman, J. D., Han, W. S., & Dentz, M. (2020). Anomalous Transport in Three-Dimensional Discrete Fracture Networks: Interplay Between Aperture Heterogeneity and Injection Modes. *Water Resources Research*, *56*(11), 1–22. https://doi.org/10.1029/2020WR027378

Karra, S., Makedonska, N., Viswanathan, H. S., Painter, S. L., & Hyman, J. D. (2015). Effect of advective flow in fractures and matrix diffusion on natural gas production. *Water Resources Research*, *51*(10), 8646–8657. https://doi.org/10.1002/2014WR016829

Kelly, J. F., Bolster, D., Meerschaert, M. M., Drummond, J. D., & Packman, A. I. (2017). FracFit: A robust parameter estimation tool for fractional calculus models. *Water Resources Research*, *53*(3), 2559–2567. https://doi.org/10.1002/2016WR019748.Received

Lee, S. H., & Kang, P. K. (2020). Three-Dimensional Vortex-Induced Reaction Hot Spots at Flow Intersections. *Physical Review Letters*, *124*(14). https://doi.org/10.1103/PhysRevLett.124.144501

Lei, Q., Latham, J. P., Tsang, C. F., Xiang, J., & Lang, P. (2015). A new approach to upscaling fracture network models while preserving geostatistical and geomechanical

characteristics. *Journal of Geophysical Research: Solid Earth*, *120*(7), 4784–4807. https://doi.org/10.1002/2014JB011736

Levy, M., & Berkowitz, B. (2003). Measurement and analysis of non-Fickian dispersion in heterogeneous porous media. *Journal of Contaminant Hydrology*, *64*(3–4), 203–226. https://doi.org/10.1016/S0169-7722(02)00204-8

Liu, H. (2017). *Fluid Flow in the Subsurface*. *Springer*. https://doi.org/10.1007/978-3-319-43449-0

Long, J. C. S., & Witherspoon, P. A. (1985). €arth Sciences Division _,.

Makedonska, N., Painter, S. L., Bui, Q. M., Gable, C. W., & Karra, S. (2015). Particle tracking approach for transport in three-dimensional discrete fracture networks: Particle tracking in 3-D DFNs. *Computational Geosciences*, *19*(5), 1123–1137. https://doi.org/10.1007/s10596-015-9525-4

Moreno, L., Tsang, C. F., Tsang, Y., & Neretnieks, I. (1990). Some anomalous features of flow and solute transport arising from fracture aperture variability. *Water Resources Research*, *26*, 2377–2391.

Neretnieks, I., Eriksen, T., & Tähtinen, P. (1982). Tracer movement in a single fissure in granitic rock: Some experimental results and their interpretation. *Water Resources Research*, *18*(4), 849–858. https://doi.org/10.1029/WR018i004p00849

Neuman, S. P. (2005). Trends, prospects and challenges in quantifying flow and transport through fractured rocks. *Hydrogeology Journal*, *13*(1), 124–147. https://doi.org/10.1007/s10040-004-0397-2

Nielsen, D. R., & Biggar, J. W. (1962). Miscible displacement: III. Theoretical considerations. *Soil Science Society Proceedings*, *26*(3), 216–221.

Nordqvist, A. W., Tsang, Y. W., Tsang, C. F., Dverstorp, B., & Andersson, J. (1992). A variable aperture fracture network model for flow and transport in fractured rocks. *Water Resources Research*, *28*(6), 1703–1713. https://doi.org/10.1029/92WR00216

Painter, S., & Cvetkovic, V. (2005). Upscaling discrete fracture network simulations: An alternative to continuum transport models. *Water Resources Research*, *41*(2), 1–10. https://doi.org/10.1029/2004WR003682

Painter, S. L., Gable, C. W., & Kelkar, S. (2012). Pathline tracing on fully unstructured control-volume grids. *Computational Geosciences*, *16*(4), 1125–1134. https://doi.org/10.1007/s10596-012-9307-1

Park, Y. J., De Dreuzy, J. R., Lee, K. K., & Berkowitz, B. (2001). Transport and intersection mixing in random fracture networks with power law length distributions. *Water Resources Research*, *37*(10), 2493–2501. https://doi.org/10.1029/2000WR000131

Pickens, F., Grisak, E., Directorate, I. W., & Canada, E. (1981). Dispersion in Hydrogeologic Systems f f (.•) d : g, *17*(6), 1701–1711.

Raats, P. A. C. (1973). *Dynamics of Fluids in Porous Media*. *Soil Science Society of America Journal* (Vol. 37). New York: Elsevier. https://doi.org/10.2136/sssaj1973.03615995003700040004x

Rajaram, H., & Gelhar, L. W. (1993). Plume scale-dependent dispersion in heterogeneous aquifers: 1. Lagrangian analysis in a stratified aquifer. *Water Resources Research*, *29*(9), 3249–3260. https://doi.org/10.1029/93WR01069

Roubinet, D., Gouze, P., Puyguiraud, A., & Dentz, M. (2022). Multi-scale random walk models for reactive transport processes in fracture-matrix systems. *Advances in Water Resources*, *164*, 104183.

Sahimi, M. (2011). *Flow and transport in porous media and fractured rock: From classical methods to modern approaches* (2nd ed.). Weinheim, Germany: Wiley-VCH.

Scheidegger, A. E. (1959). An evaluation of the accuracy of the diffusivity equation for describing miscible displacement in porous media. In *Theory of Fluid Flow in Porous Media Conference* (pp. 101–116). Retrieved from https://www.amazon.com/evaluation-accuracy-diffusivity-describing-displacement/dp/B0007JVSE6

Sherman, T., Fakhari, A., Miller, S., Singha, K., & Bolster, D. (2017). Parameterizing the spatial Markov model from breakthrough curve data alone. *Water Resources Research*, *53*, 10888–10898.

Sherman, Thomas, Hyman, J. D., Bolster, D., Makedonska, N., & Srinivasan, G. (2019). Characterizing the impact of particle behavior at fracture intersections in three-dimensional discrete fracture networks. *Physical Review E*, *99*(1). https://doi.org/10.1103/PhysRevE.99.013110

Silliman, S. E., & Simpson, E. S. (1987). Laboratory evidence of the scale effect in dispersion of solutes in porous media. *Water Resources Research*, *23*(8), 1667–1673. https://doi.org/10.1029/WR023i008p01667

Vermilye, J. M., & Scholz, C. H. (1995). Relation between vein length and aperture. *Journal of Structural Geology*, *17*(3), 423–434. https://doi.org/10.1016/0191-8141(94)00058-8

Wood, A. T. A. (1994). Simulation of the von mises fisher distribution. *Communications in Statistics - Simulation and Computation*, *23*(1), 157–164. https://doi.org/10.1080/03610919408813161

Yuan, B., & Wood, D. A. (2018). A holistic review of geosystem damage during unconventional oil, gas and geothermal energy recovery. *Fuel*, *227*, 99–110.

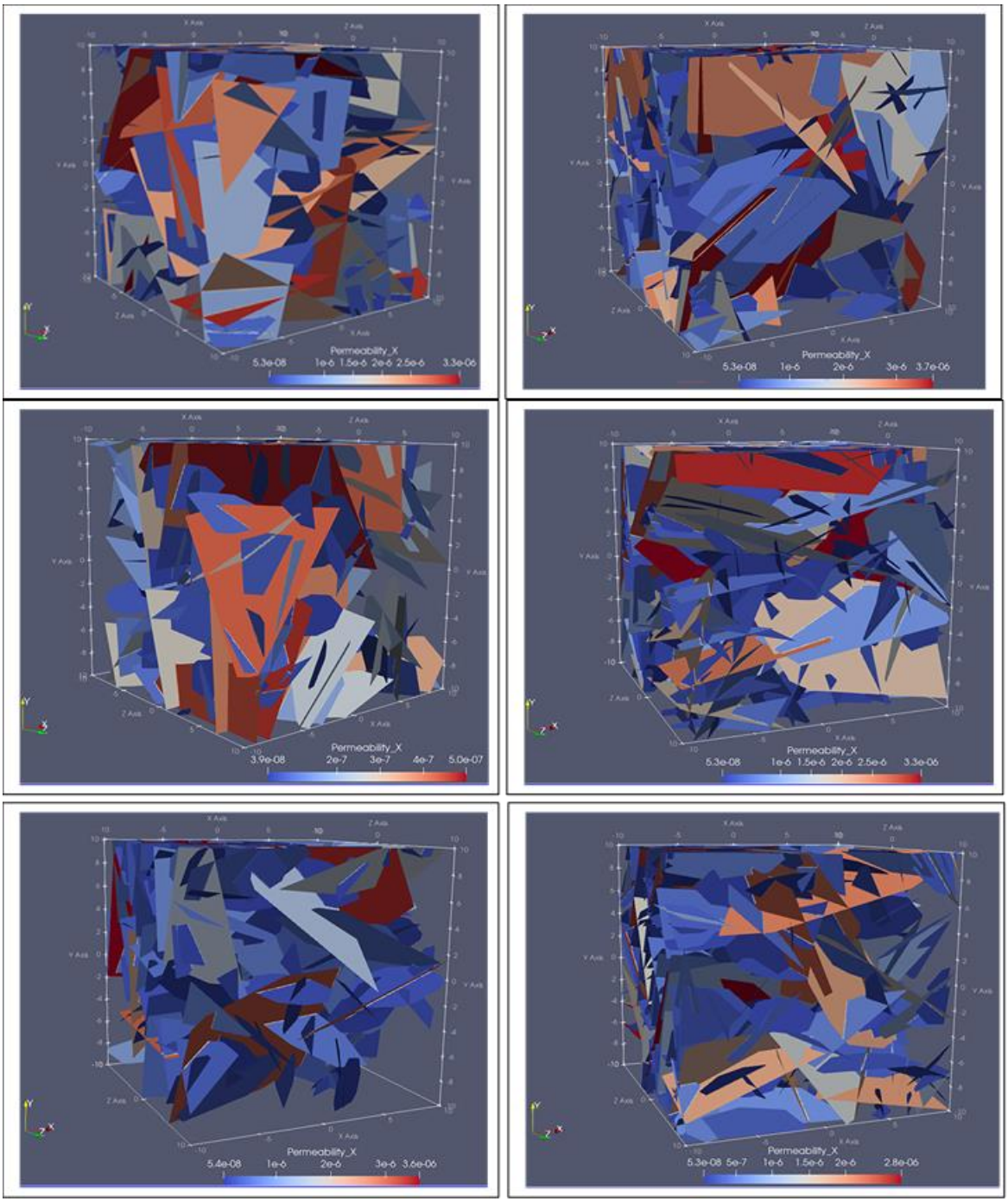
Zhu, W., Khirevich, S., & Patzek, T. W. (2021). Impact of Fracture Geometry and Topology on the Connectivity and Flow Properties of Stochastic Fracture Networks. *Water Resources Research*, *57*(7). https://doi.org/10.1029/2020WR028652

Zimmerman, R. W., & Bodvarsson, G. S. (1996). Hydraulic conductivity of rock fractures. *Transport in Porous Media*, *23*(1), 1–30. https://doi.org/10.1007/BF00145263

Zoback, M. D., & Kohli, A. H. (2019). *Unconventional Reservoir Geomechanics. Cambridge University Press.*

Zyvoloski, G. A., & Vesselinov, V. V. (2006). An investigation of numerical grid effects in parameter estimation. *Ground Water*, *44*(6), 814–825. https://doi.org/10.1111/j.1745-6584.2006.00203.x

Appendix A - DFN permeability field and standard deviation for CTRW model parameters



Appendix A Figure A.1. Discrete fracture networks of L = 20 m showing the imapacts of TPL $\boldsymbol{\alpha}$ on the permeability of the network: left p30 = 0.05 (top: $\boldsymbol{\alpha}$ = 1.5), (middle: $\boldsymbol{\alpha}$ = 2.0), (bottom: $\boldsymbol{\alpha}$ = 2.5). Right p30 = 0.10 (top: $\boldsymbol{\alpha}$ = 1.5), (middle: $\boldsymbol{\alpha}$ = 2.0), (bottom: $\boldsymbol{\alpha}$ = 2.5). Note that colors indicate the permeability of individual fractures and their size in the network, with larger fractures having warmer colors.

Appendix A Table A.1. Network characterization (Density and Topology), and standard deviation values showing the variability between the ensembled CTRW optimized parameters.

Fracture Network Topological Properties						**CTRW Model Parameters (Standard deviation)**				
P30	**L(m)**	**P32**	N_f	α	K **(m^2)**	β	$v(\mathbf{ms^{-1}})$	$D(\mathbf{m^2s^{-1}})$	$t_1(\mathbf{s})$	$t_2(\mathbf{s})$
0.05	20	1.30	193	1.5	1.35×10^{-09}	0.21	1.84×10^{01}	6.00×10^{01}	3.96×10^{-02}	1.24×10^{04}
		1.17	182	2.0	6.58×10^{-11}	0.20	5.91×10^{00}	2.15×10^{00}	2.54×10^{-01}	9.35×10^{03}
		1.03	166	2.5	5.09×10^{-10}	0.18	3.90×10^{01}	6.05×10^{01}	3.41×10^{-02}	7.67×10^{03}
	35	1.33	1011	1.5	1.09×10^{-09}	0.15	4.41×10^{01}	5.37×10^{01}	1.95×10^{-02}	5.97×10^{02}
		1.28	993	2.0	6.52×10^{-10}	0.11	1.35×10^{02}	1.00×10^{02}	4.85×10^{-03}	4.73×10^{02}
		1.21	974	2.5	3.82×10^{-10}	0.07	1.33×10^{02}	1.60×10^{02}	2.94×10^{-02}	1.67×10^{02}
	50	1.44	3124	1.5	1.50×10^{-09}	0.18	1.78×10^{00}	1.07×10^{01}	3.13×10^{-01}	6.23×10^{02}
		1.40	3153	2.0	7.39×10^{-10}	0.21	2.04×10^{00}	7.87×10^{00}	2.04×10^{-01}	2.65×10^{03}
		1.34	3139	2.5	4.28×10^{-10}	0.16	6.19×10^{00}	1.43×10^{01}	2.36×10^{-02}	2.34×10^{03}
0.10	20	1.77	338	1.5	2.21×10^{-09}	0.09	3.90×10^{02}	8.14×10^{01}	3.20×10^{-04}	1.70×10^{01}
		1.74	342	2.0	1.56×10^{-09}	0.19	2.53×10^{01}	4.44×10^{01}	2.62×10^{-02}	3.57×10^{00}
		1.65	338	2.5	1.17×10^{-09}	0.12	1.17×10^{01}	4.62×10^{01}	8.03×10^{-03}	6.36×10^{02}
	35	1.75	1648	1.5	1.71×10^{-09}	0.03	3.38×10^{02}	2.05×10^{02}	4.42×10^{-05}	1.62×10^{01}
		1.70	1679	2.0	1.05×10^{-09}	0.19	6.50×10^{01}	2.32×10^{01}	3.02×10^{-02}	3.08×10^{02}
		1.69	1676	2.5	9.32×10^{-10}	0.15	7.27×10^{01}	1.52×10^{02}	1.70×10^{-02}	1.70×10^{02}

		1.88	4972	1.5	1.79×10^{-09}	0.03	3.40×10^{01}	7.65×10^{01}	5.67×10^{-04}	3.43×10^{02}
	50	1.86	5052	2.0	1.38×10^{-09}	0.04	4.95×10^{01}	8.63×10^{01}	8.67×10^{-04}	8.08×10^{02}
		1.82	5107	2.5	9.71×10^{-10}	0.15	1.23×10^{00}	4.59×10^{00}	3.26×10^{-01}	6.05×10^{02}

p30: fracture density, L: domain size, N_f: final number of fractures after isolated fractures have been removed from network, p32: initial fracture surface area, α: truncated power-law exponent for fracture length distribution, K: system permeability; β: CTRW-TPL exponent, v: solute velocity, D: dispersion, t_1: transition time, t_2: cut-off time.

Appendix B - Scripts for generating fracture networks, running flow and transport simulations

Discrete Fracture Network Generation Script

```
//=================================================================
==========
// General Options & Fracture Network Parameters:


stopCondition: 0
   /* 0: stop once nPoly fractures are accepted (Defined below)
      1: stop once all family's p32 values are equal or greater than the families
            target p32 values (defined in stochastic family sections)
   */


nPoly: 4288
    /* Used when stopCondition = 0
       Total number of fractures you would like to have
       in the domain you defined. The program will complete
       once you have nPoly number of fractures,
       maxPoly number of polygon/fracture rejections,
       rejPoly number of rejections in a row, or reach a
       specified fracture cluster size if using
       stoppingParameter = -largestSize  */
```

```
outputAllRadii: 0
/* 0: Do not output all radii file.
   1: Include file of all raddii (accepted+rejected fractures)
      in output files.
*/


domainSize: {35,35,35}
     /* Mandatory Parameter.
        Creates a domain with dimension x*y*z centered at the origin.*/

numOfLayers: 0    //number of layers
layers:
{0}

/*  Layers need to be listed line by line
    Format: {minZ, maxZ}

    The first layer listed is layer 1, the second is layer 2, etc
    Stochastic families can be assigned to theses layers (see stochastic
    shape family section)
```

```
*/


numOfRegions: 0 // Number of regions
regions:
{}

/*  Regions need to be listed line by line
    Format: {minX, maxX, minY, maxY, minZ, maxZ}

    The first region listed is region 1, the second is region 2, etc
    Stochastic families can be assigned to theses layers (see stochastic
    shape family section)
*/


orientationOption: 0
/* Fracture Orientation Option
    0 : Spherical Coordinates
    1 : Trend / Plunge
    2 : Dip / Strike
*/
```

h: 0.15

/* Minimum fracture length scale(meters)

Any fracture with a feature, such as and intersection, of less than h will be rejected. */

//==========

======//

/* Fracture Network Parameters: */

tripleIntersections: 0

/* Options: 0: Off

1: On */

printRejectReasons: 0

/* Useful for debugging,

This option will print all fracture rejection reasons as they occur.

0: disable

1: print all rejection reasons to screen

*/

disableFram: 0

visualizationMode: 0

/* Options: 0 or 1

Used during meshing:

0: creates a fine mesh, according to h parameter;

1: produce only first round of triangulations. In this case no

modeling of flow and transport is possible. */

seed: 0

//Seed for random generator. 0 = seed off clock

domainSizeIncrease: {8, 8, 8}

//temporary size increase for inserting fracture centers outside domain

//increases the entire width by this amount. So, {1,1,1} will increase

//the domain by adding .5 to the +x, and subbtracting .5 to the -x, etc

keepOnlyLargestCluster: 1

/* 0 = Keep any clusters which connects the specified

boundary faces in boundaryFaces option below

1 = Keep only the largest cluster which connects

the specified boundary faces in boundaryFaces option below

```
   */

ignoreBoundaryFaces: 0
/*
    0 = use boundaryFaces option below

    1 = ignore boundaryFaces option and keep all clusters and
    will still remove fractures with no intersections
*/


boundaryFaces: {1,1,0,0,0,0}
/*  DFN will only keep clusters with connections to domain boundaries which are set to 1:

   boundaryFaces[0] = +X domain boundary
   boundaryFaces[1] = -X domain boundary
   boundaryFaces[2] = +Y domain boundary
   boundaryFaces[3] = -Y domain boundary
   boundaryFaces[4] = +Z domain boundary
   boundaryFaces[5] = -Z domain boundary

   Be sure to set ignoreBoundaryFaces to 0 when using this feature.
*/
```

```
rejectsPerFracture: 10  /*If fracture is rejected, it will be re-translated to a new
                        position this number of times.

                        This helps hit distribution targets for stochastic families
                        (Set to 1 to ignore this feature)    */


outputAcceptedRadiiPerFamily: 0


//=====================================================================
=======
//                Shape and Probability Parameters
//=====================================================================
=======

//user rectangles and user Ellipses defined in their cooresponding files

famProb: {1}
/* Probability of occurrence of each family of randomly distrubuted rectangles
   and ellipses.
```

User-ellipses and user-rectangles insertion will be attempted with 100% likelihood, but with possibility they may be rejected.

The famProb elements should add up to 1.0 (for %100).

The probabilities are listed in order of families starting with all stochastic ellipses, and then all stochastic rectangles.

For example:

If then there are two ellipse families, each with probabiliy .3,

and two rectangle families, each with probabiliy .2, famProb will be:

famProb: {.3,.3,.2,.2} Notice: famProb elements add to 1

*/

insertUserRectanglesFirst: 0

/*

1: User rectangles will be inserted first

0: User ellipses will be inserted first

*/

/*==========

========*/

//==========

=======

```
//                  Elliptical Fracture Options
//      NOTE: Number of elements must match number of ellipse families
//           (first number in nShape input parameter)
//=============================================================================
=======
/*=============================================================================
========*/


//Number of ellipse families
nFamEll: 1
   //Having this option = 0 will ignore all rectangle family variables


eLayer: {0}
   /* Defines which domain the family belongs to.
    Layer 0 is the entire domain.
    Layers numbered > 0 corresponds to layers defined above
    1 corresponds to the first layer listed, 2 is the next layer listed, etc */



eRegion: {0}
   /* Defines which domain the family belongs to.
    Region 0 is the entire domain.
    Regions numbered > 0 correspond to layers defined above
```

1 correspond to the first region listed, 2 is the next region listed, etc */

//edist is a mandatory parameter if using statistically generated ellipses

edistr: {2} /* Ellipse statistical distribution options:

1 - lognormal distribution

2 - truncated power law distribution

3 - exponential distribution

4 - constant */

ebetaDistribution: {1} /* Beta is the rotation around the polygon's normal

vector, with the polygon centered on x-y plane at the origin

0 - uniform distribution [0, 2PI]

1 - constant angle (specified below by "ebeta") */

e_p32Targets: {0.5}

/* Elliptical families target fracture intensity per family.

When using stopCondition = 1 (defined at the top of the input file), families will

be inserted until the families desired fracture intensity has been reached.

Once all families desired fracture intensity has been met, fracture generation will

be complete.

*/

//===

=======

// Parameters used by all stochastic ellipse families

// Mandatory Parameters if using statistically generated ellipses

easpect: {2} /* Aspect ratio. Used for lognormal and truncated

power law distribution. */

enumPoints: {8} /*Number of vertices used in creating each elliptical

fracture family. Number of elements must match number

of ellipse families (first number in nShape) */

eAngleOption: 1 /* All angles for ellipses:

0 - degrees

0 - Radians (Must use numerical value for PI) */

etheta: {0} /*Ellipse fracture orientation.

The angle the normal vector makes with the z-axis */

ephi: {0} /* Ellipse fracture orientation.

The angle the projection of the normal onto the x-y plane

makes with the x-axis */

ebeta: {90} /* rotation around the normal vector */

ekappa: {1} /*Parameter for the fisher distributions. The bigger, the more

similar (less diverging) are the elliptical family's

normal vectors */

//===

=======

// Options Specific For Ellipse Lognormal Distribution (edistr=1):

// Mandatory Parameters if using ellispes with lognormal distribution

// NOTE: Number of elements must match number of

// ellipse families (first number in nShape)

eLogMean: {2} //Mean value For Lognormal Distribution.

esd: {.5} // Standard deviation for lognormal distributions of ellipses

eLogMin: {1}

eLogMax: {15}

//===

=======

// Options Specific For Ellipse Exponential Distribution (edistr=3):

// Mandatory Parameters if using ellispes with exponential distribution

eExpMean: {2} //Mean value for Exponential Distribution

eExpMin: {1}

eExpMax: {5}

//===

=======

// Options Specific For Constant Size of ellipses (edistr=4):

econst: {10, 10, 10} // Constant radius, defined per family

```
//=================================================================
=======
// Options Specific For Ellipse Truncated Power-Law Distribution (edistr=2)
// Mandatory Parameters if using ellipses with truncated power-law dist.

// NOTE: Number of elements must match number
//       of ellipse families (first number in nShape)

emin: {1.5} // Minimum radius for each ellipse family.
            // For power law distributions.

emax: {6.75}  // Maximum radius for each ellipse family.
              // For power law distributions.

ealpha: {1.5} // Alpha. Used in truncated power-law
                    // distribution calculation


/*=================================================================
==============*/
/*=================================================================
==============*/
/*                      Rctangular Fractures Options                */
```

```
/* NOTE: Number of elements must match number of rectangle families              */
/*       (second number in nShape parameter)                                     */
/*=======================================================================
==============*/
/*=======================================================================
==============*/

//Number of rectangle families
nFamRect: 0
   //Having this option = 0 will ignore all rectangle family variables



rLayer: {}
   /* Defines which domain the family belongs to.
    Layer 0 is the entire domain.
    Layers numbered > 0 corresponds to layers defined above
    1 corresponds to the first layer listed, 2 is the next layer listed, etc */



rRegion: {}
   /* Defines which domain the family belongs to.
    Region 0 is the entire domain.
    Regions numbered > 0 correspond to layers defined above
```

1 corresponds to the first region listed, 2 is the next region listed, etc */

/*rdist is a mandatory parameter if using statistically generated rectangles */

rdistr: {} /* Rectangle statistical distribution options:

1 - lognormal distribution

2 - truncated power law distribution

3 - exponential distribution

4 - constant */

rbetaDistribution: {} /* Beta is the rotation/twist about the z axis

with the polygon centered on x-y plane at the origin

before rotation into 3d space

0 - uniform distribution [0, 2PI]

1 - constant angle (specified below by "rbeta")

*/

r_p32Targets: {}

/* Rectangle families target fracture intensity per family.

When using stopCondition = 1 (defined at the top of the input file), families will

be inserted untill the families desired fracture intensity has been reached.

```
Once all families desired fracture intensity has been met, fracture generation will
be complete.
*/

//=================================================================
========
// Parameters used by all stochastic rectangle families
// Mandatory Parameters if using statistically generated rectangles

raspect: {}  /* Aspect ratio */

rAngleOption: 0     /* All angles for rectangles:
                    0 - degrees
                    1 - radians (must be numerical value, cannot use "Pi") */

rtheta: {0, 1.57, 1.57} /*Rectangle fracture orientation.
                      The angle the normal vector makes with the z-axis */

rphi: {0, 0, 1.57} /* Rectangle fracture orientation.
                The angle the projection of the normal onto the x-y
                plane makes with the x-axis */

rbeta: {0, 0, 0}   /* rotation around the normal vector */
```

```
rkappa: {100, 100, 100}  /*Parameter for the fisher distributions. The bigger,
                         the more similar (less diverging) are the rectangle
                         family's normal vectors  */

//=================================================================
=========
// Options Specific For Rectangle Lognormal Distribution (rdistr=1):
// Mandatory Parameters if using rectangles with lognormal distribution

rLogMean: {1.6}   /*For Lognormal Distribution.
                  Mean radius (1/2 rectangle length) in
                  lognormal distribution for rectangles. */


rsd: {.4}     /* Standard deviation for lognormal distributions of
                   rectangles */

rLogMin: {2}

rLogMax: {10}
```

```
//=================================================================
=========
// Options Specific For Rectangle Truncated Power-Law Distribution (rdistr=2):
// Mandatory Parameters if using rectangles with power-law distribution

 rmin: {1,1}         /* Minimum radius for each rectangle family.
                        For power law distributions. */

 rmax: {6,5}    /* Maximum radius for each rectangle family.
                     For power law distributions. */

 ralpha: {2.4,2.5}    // Alpha. Used in truncated power-law
                  // distribution calculation


/*=================================================================
========*/
/* Options Specific For Rectangle Exponential Distribution (edistr=3):       */
/* Mandatory Parameters if using rectangles with exponential distribution   */

rExpMean: {2}  //Mean value for Exponential  Distribution

rExpMin: {1}
```

```
rExpMax: {10}

/*================================================================
========*/
/* Options Specific For Constant Size of rectangles (edistr=4):          */

rconst: {2, 2, 2}  // Constant radius, defined per rectangular family



/*================================================================
========*/
/*================================================================
========*/
/*  User-Specified Ellipses                                      */
/*  Mandatory Parameters if using user-ellipses                  */
/*  NOTE: Number of elements must match number of user-ellipse families.    */
/*  NOTE: Only one user-ellipse is placed into the domain per defined       */
/*        user-ellipse, with possibility of being rejected                  */
/*================================================================
========*/
/*================================================================
========*/
```

userEllipsesOnOff: 0

/* 0 - User ellipses off

    1 - User ellipses on

*/

UserEll_Input_File_Path:

/home/jharrod/GitProjects/DFNGen/DFNC++Version/inputFiles/userPolygons/uEllInput.dat

/*==========

========*/

/*==========

========*/

/* User-Specified Ellipses   */

/* Mandatory Parameters if using user-ellipses   */

/* NOTE: Number of elements must match number of user-ellipse families.   */

/* NOTE: Only one user-ellipse is placed into the domain per defined   */

/*   user-ellipse, with possibility of being rejected   */

/*==========

========*/

/*==========

========*/

userEllByCoord: 0

/* 0 - User ellipses defined by coordinates off

    1 - User ellipses defined by coordinates on

*/

EllByCoord_Input_File_Path:

/home/jharrod/GitProjects/DFNGen/DFNC++Version/inputFiles/userPolygons/ellCoords.dat

/*==

========*/

/*==

========*/

/* User-Specified Rectangles   */

/* NOTE: Number of elements must match number of user-ellipse families   */

/* NOTE: Only one user-rectangle is placed into the domain per defined   */

/*   user-rectangle, with possibility of being rejected   */

/*==

========*/

/*==

========*/

userRectanglesOnOff: 0

/* 0 - User Rectangles off

    1 - User Rectangles on

*/

UserRect_Input_File_Path:

/home/jharrod/GitProjects/DFNGen/DFNC++Version/inputFiles/userPolygons/ignoreConnTest.dat

/*==========

========*/

/*==========

========*/

/*   */

/* User Rectangles Defined By Coordinates   */

/*   */

/*==========

========*/

/*==========

========*/

userRecByCoord: 0

```
/*  0 - User defined rectangles by coordinates off
    1 - User defined rectangles by coordinates on
*/

RectByCoord_Input_File_Path:
~/GitProjects/DFNGen/DFNC++Version/inputFiles/userPolygons/rectCoords.dat

/*  WARNING: This option can cause LaGriT errors if the polygon
            vertices are not put in clockwise or counter-clockwise order.
            If errors (Usualy seg fualt during meshing in LaGriT),
            make sure the vertices are in clockwise or counter clockwise
            order. Also, coordinates must be co-planar.
*/


/*==========================================================================
========*/
// Aperture [m]
/* Mandatory parameter, and can be specified in several ways:
- 1)meanAperture and stdAperture for using LogNormal distribution.
- 2)apertureFromTransmissivity, first transmissivity is defined, and then,
  using a cubic law, the aperture is calculated;
- 3)constantAperture, all fractures, regardless of their size, will have
```

```
  the same aperture value;
- 4)lengthCorrelatedAperture, aperture is defined as a function of fracture size*/

//NOTE: Only one aperture type may be used at a time

aperture: 4  //choice of aperture option described above

//(**** 1)meanAperture and stdAperture for using LogNormal distribution.********)
meanAperture:  3 /*Mean value for aperture using
                          normal distribution */
stdAperture: 0.8  //Standard deviation

/*(****** 2)apertureFromTransmissivity, first transmissivity is defined,
  and then, using a cubic law, the aperture is calculated;***************/
apertureFromTransmissivity: {1.6e-9, 0.8}
    /* Transmissivity is calculated as transmissivity = F*R^k,
       where F is a first element in aperturefromTransmissivity,
       k is a second element and R is a mean radius of a polygon.
       Aperture is calculated according to cubic law as
       b=(transmissivity*12)^1/3 */

/*(****** 3)constantAperture, all fractures, regardless of their size,
   will have the same aperture value;    **********************************/
```

```
constantAperture: 1e-5  //Sets constant aperture for all fractures

/*(******** 4)lengthCorrelatedAperture, aperture is defined as a function of
      fracture size ********************/

lengthCorrelatedAperture: {0.0001694, 1.41}
   /*Length Correlated Aperture Option:
     Aperture is calculated by: b=F*R^k,
     where F is a first element in lengthCorrelatedAperture,
     k is a second element and R is a mean radius of a polygon.*/


//=================================================================
========
//Permeability
/* Options:
   0: Permeability of each fracture is a function of fracture aperture,
    given by k=(b^2)/12, where b is an aperture and k is permeability
   1: Constant permeabilty for all fractures */

permOption: 0  //See above for options
```

constantPermeability: 1e-12 //Constant permeability for all fractures

forceLargeFractures: 0

outputFinalRadiiPerFamily: 0

radiiListIncrease: 0.1

removeFracturesLessThan: 0

keepIsolatedFractures: 0

/* 0 - Remove any isolated fracture (not clusters)

    1 - Keep all fractures in the domain

*/

/*==========

========*/

/*==========

========*/

/*   */

/* User Polygon Defined By Coordinates   */

/*   */

```
/*=================================================================
========*/
/*=================================================================
========*/

userPolygonByCoord: 0
/*  0 - User defined polygon by coordinates off
    1 - User defined polygon by coordinates on
*/

PolygonByCoord_Input_File_Path: ./

/*WARNING: userDefCoordRec can cause LaGriT errors because the polygon
vertices are not put in clockwise or counter-clockwise order.
If errors (Usualy seg fualt during meshing if using LaGriT),
try to reorder the points till u get it right.
Also, coordinates must be co-planar */
```

Flow Simulation Script

```
# Jan 13, 2014
# Nataliia Makedonska, Satish Karra, LANL
#=============================================

SIMULATION
```

```
  SIMULATION_TYPE SUBSURFACE
  PROCESS_MODELS
    SUBSURFACE_FLOW flow
      MODE RICHARDS
    /
  /
END
SUBSURFACE


DFN


#========================== discretization
==================================
GRID
  TYPE unstructured_explicit full_mesh_vol_area.uge
  GRAVITY 0.d0 0.d0 0.d0
END



#========================== fluid properties
================================
FLUID_PROPERTY
  DIFFUSION_COEFFICIENT 1.d-9
```

```
END

DATASET Permeability
  FILENAME dfn_properties.h5
END

#========================== material properties
============================
MATERIAL_PROPERTY soil1
  ID 1
  POROSITY 0.25d0
  TORTUOSITY 0.5d0
  CHARACTERISTIC_CURVES default
  PERMEABILITY
    DATASET Permeability
  /
END


#========================== characteristic curves
============================
CHARACTERISTIC_CURVES default
  SATURATION_FUNCTION VAN_GENUCHTEN
```

```
    M 0.5d0
    ALPHA  1.d-4
    LIQUID_RESIDUAL_SATURATION 0.1d0
    MAX_CAPILLARY_PRESSURE 1.d8
  /
  PERMEABILITY_FUNCTION MUALEM_VG_LIQ
    M 0.5d0
    LIQUID_RESIDUAL_SATURATION 0.1d0
  /
END

#========================== output options
=================================
OUTPUT
  TIMES s 0.01 0.05 0.1 0.2 0.5 1
#  FORMAT TECPLOT BLOCK
  PRINT_PRIMAL_GRID
  FORMAT VTK
  MASS_FLOWRATE
  MASS_BALANCE
  VARIABLES
    LIQUID_PRESSURE
    PERMEABILITY
```

```
  /
END

#========================== times
=========================================
TIME
  INITIAL_TIMESTEP_SIZE  1.d-8 s
  FINAL_TIME 1.d0 d
  MAXIMUM_TIMESTEP_SIZE 10.d0 d
  STEADY_STATE
END

# REFERENCE_PRESSURE 1500000.

#========================== regions
=========================================
REGION All
  COORDINATES
    -1.d20 -1.d20 -1.d20
    1.d20 1.d20 1.d20
  /
END
```

```
REGION inflow
  FILE pboundary_left_w.ex
END

REGION outflow
  FILE pboundary_right_e.ex
END

#========================== flow conditions
=================================
FLOW_CONDITION initial
  TYPE
    PRESSURE dirichlet
  /
  PRESSURE 1.01325d6
END


FLOW_CONDITION outflow
  TYPE
    PRESSURE dirichlet
  /
  PRESSURE 1.d6
```

```
END

FLOW_CONDITION inflow
  TYPE
    PRESSURE dirichlet
  /
  PRESSURE 2.d6
END

#========================== condition couplers
==============================
# initial condition
INITIAL_CONDITION
  FLOW_CONDITION initial
  REGION All
END


BOUNDARY_CONDITION INFLOW
  FLOW_CONDITION inflow
  REGION inflow
END
```

```
BOUNDARY_CONDITION OUTFLOW
  FLOW_CONDITION outflow
  REGION outflow
END

#========================== stratigraphy couplers
============================
STRATA
  REGION All
  MATERIAL soil1
END

END_SUBSURFACE
```

Solute Transport Simulation Script

```
/*************************************************************************/
/*   CONTROL FILE FOR PARTICLE TRACKING IN DISCRETE FRACTURE NETWORK
*/
/*************************************************************************/

/**********************  INPUT FILES: grid ****************************/
/**** input files with grid of DFN, mainly its output of DFNGen ******/
param: params.txt
poly: poly_info.dat
```

inp: full_mesh.inp

stor: full_mesh.stor

boundary: allboundaries.zone

/* boundary conditions: reading the nodes that belong to in-flow and

out-flow boundaries. Should be consistent with those applied to obtain

steady state pressure solution (PFLOTRAN) */

/*1 - top; 2 - bottom; 3 - left_w; 4 - front_s; 5 - right_e; 6 - back_n */

in-flow-boundary: 3

out-flow-boundary: 5

/********** INPUT FILES: PFLOTRAN flow solution **********/

PFLOTRAN: yes

PFLOTRAN_vel: darcyvel.dat

PFLOTRAN_cell: cellinfo.dat

PFLOTRAN_uge: full_mesh_vol_area.uge

/********** INPUT FILES: FEHM flow solution **********/

/*currently we are using PFLOTRAN , but the code would work with FEHM, too */

FEHM: no

FEHM_fin: tri_frac.fin

/********** OUTPUT FILES **********/

/* initial grid info structure output, usefull for debugging */

out_grid: no

/* flow field: 3D Darcy velocities: output file has an each nodes position

and its Darcy velocity, reconstructed from fluxes */

out_3dflow: no

/* out initial positions of particles into separate file */

out_init: no

/* out particle trajectories tortuosity file, torts.dat */

out_tort: no

/*************** output options for particles trajectories ***************/

/* output frequency is set according to trajectories curvature. We check the

curvature of particles trajectory each segment, from intersection to intersection.

If it's like a straight line, then the output is less frequent (in case of

"out_curv:yes", if "no", the output file will contain every time step) */

out_curv: yes

/* output into avs file (GMV visualization, Paraview visualization) */

out_avs: no

/* output into trajectories ascii files (veloc+posit+cell+fract+time) */

out_traj: no

out_fract: no

/* temporary outputs (every time step from intersection to intersection)*/

```
/* use outputs to file or memory buffer. Memory buffer by default */
out_filetemp: no

/************* output directories ***********************************/

out_dir: dfnTrans_output /* path and name of directory where all the particle
                tracking results will be written*/


out_path: trajectories /*name of directory where all particle
                trajectories will be saved, in out_dir path */

/* name of resultant file (in out_dir path), which contains total travel time and
                                final positions of particles */
out_time: partime.dat

/**************** Intersection Mixing Rule ********************************/
/****streamline_routing: if yes - streamline routing is the selected subgrid process
    otherwise the complete mixing rule is selected ****/
streamline_routing: no

/***************** PARTICLES INITIAL POSITIONS ******************************/
/****init_nf: if yes - the same number of particles (init_partn) will be placed
```

```
    on every boundary fracture edge on in-flow boundary,
    equidistant from each other ****/
init_nf: no
init_partn: 10

/****init_eqd: if yes - particles will be placed on the same distance from
    each other on all over in-flow boundary edges **********************/
init_eqd: no  //maximum number of particles that user expects on one boundary edge
init_npart: 100

/*** all particles start from the same region at in-flow boundary, in a range
   {in_xmin, in_xmax,in_ymin, in_ymax, in_zmin, in_zmax} **************/
init_oneregion: no
in_partn: 100000
in_xmin: -50.0
in_xmax: -50.0
in_ymin: -20.0
in_ymax:  20.0
in_zmin: -15.0
in_zmax:  0.0

/**** all particles are placed randomly over all fracture surface
    (not only on boundary edges!) ***********************************/
```

init_random: no

// total number of particles

in_randpart: 100

init_matrix: no

/***** particles positions distributed according to in-flow flux weight *********/

init_fluxw: yes

init_totalnumber: 50000

/****************** FLOW AND FRACTURE PARAMETERS **********************/

porosity: 1.0 // porosity

density: 997.73 //fluid density

satur: 1.0

thickness: 1.0 //DFN aperture (used in case of no aperture file provided)

/************************ APERTURE *********************************/

aperture: yes //DFN aperture

aperture_type: frac //aperture is giving per cell (type "cell")

//   or per fracture (type "frac")

// for now we use an aperture giving per fracture

aperture_file: aperture.dat

/************* TIME DOMAIN RANDOM WALK ******************************/

tdrw: no

tdrw_porosity: 0.02

tdrw_diffcoeff: 1.0e-5

/********************* TIME **/

timesteps: 2000000

//units of time (years, days, hours, minutes)

time_units: years

/**** flux weighted particles*/

/**** in case of random initial positions of particles - it's aperture weighted **/

flux_weight: yes

/* random generator seed */

seed: 337799

/********************** Control Plane/Cylinder Output ********************/

/*** virtual Control planes will be built in the direction of flow.

Once particle crosses the control plane, it's position, velocity, time

will output to an ascii file. ****/

ControlPlane: no

/* the path and directory name with all particles output files */

control_out: outcontroldir

/* Delta Control Plane - the distance between control planes */

delta_Control: 1

/* ControlPlane: direction of flow: x-0; y-1; z-2 */

flowdir: 0

/**/

/endendend/

END