AGU100 ADVANCING EARTH AND SPACE SCIENCE

Water Resources Research

RESEARCH ARTICLE

10.1029/2017WR022368

Key Points:
- Graphs provide an elegant way to interrogate and characterize fracture networks
- Graphs combined with discrete fracture network models are a powerful combination for flow and transport simulations in fractured rocks
- The choice of mapping from a discrete fracture network model to graph should be chosen based on the quantity of interest

Correspondence to:
H. S. Viswanathan,
viswana@lanl.gov

Citation:
Viswanathan, H. S., Hyman, J. D., Karra, S., O'Malley, D., Srinivasan, S., Hagberg, A., & Srinivasan, G. (2018). Advancing graph-based algorithms for predicting flow and transport in fractured rock. *Water Resources Research*, *54*. https://doi.org/10.1029/2017WR022368

Received 5 DEC 2017
Accepted 18 JUL 2018
Accepted article online 31 JUL 2018

Advancing Graph-Based Algorithms for Predicting Flow and Transport in Fractured Rock

H. S. Viswanathan[1], J. D. Hyman[1], S. Karra[1], D. O'Malley[1], S. Srinivasan[1,2], A. Hagberg[3], and G. Srinivasan[4]

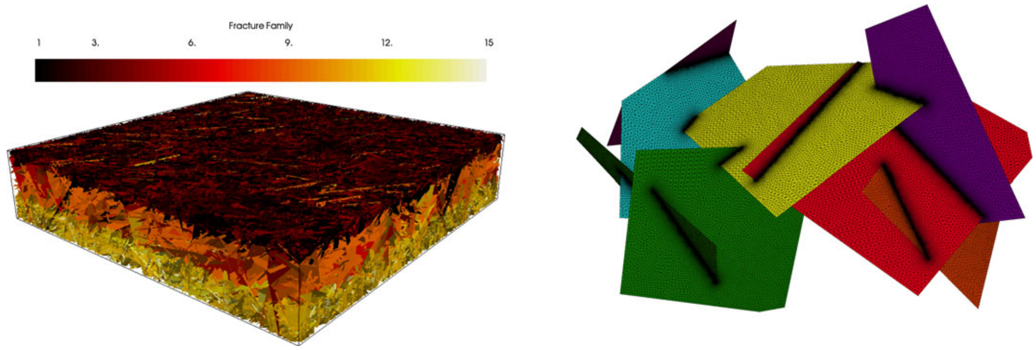
[1]Computational Earth Science (EES-16), Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, USA, [2]Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA, [3]Computer, Computational and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, USA, [4]Verification and Analysis (XCP-8), X Computational Physics Division, Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract Discrete fracture network (DFN) models are a powerful alternative to continuum models for subsurface flow and transport simulations because they explicitly include fracture geometry and network topology, thereby allowing for better characterization of the latter's influence on flow and transport through fractured media. Recent advances in high performance computing have opened the door for flow and transport simulations in large explicit three-dimensional DFN, but this increase in model fidelity and system size comes at a huge computational cost because of the large number of mesh elements required to represent thousands of fractures (with sizes that can range several orders of magnitude, from millimeter to kilometer). In most subsurface applications, fracture characteristics are only known statistically and numerous realizations are needed to bound uncertainty in flow and transport in the system, thereby exacerbating the computational burden. Graphs provide a simple and elegant way to characterize, query, and interrogate fracture network connectivity. We propose a DFN model reduction framework where various graph representations are used in conjunction with high-fidelity DFN simulations to increase computational efficiency while retaining accuracy of key quantities of interest. The appropriate choice of a graph representation, namely, which attributes of the DFN are to be represented as nodes and which ones as edges connecting those nodes, depends on the relevant scientific questions. We demonstrate that the proposed DFN model reduction framework provides an efficient means for DFN modeling through both system reduction of the DFN using graph-based properties and combining DFN and graph-based flow and transport simulations.

1. Introduction

Discrete fracture network (DFN) models incorporate the important effects of fracture structure and network topology to better characterize their influence on flow and transport through fractured systems. In the DFN methodology, each fracture in the network is assigned a shape, location, aperture, and orientation by sampling distributions whose parameters are based on site characterization. Until recently, DFN models were limited to one-dimensional pipe network approximations (Cacas, 1990; Dershowitz & Fidelibus, 1999), two-dimensional systems (de Dreuzy et al., 2001, 2002, 2004), or relatively small three-dimensional systems (Bogdanov et al., 2007). However, recent advances in high performance computing have opened the door for flow and transport simulations in large explicit three-dimensional discrete fracture networks (Berrone et al., 2013, 2015; Erhel et al., 2009; Hyman, Karra, et al., 2015; Hyman, Painter, et al., 2015; Joyce et al., 2014; Mustapha & Mustapha, 2007; Pichot et al., 2010, 2012). In these methods, a mesh representation of each fracture in the network is created, and the flow equations are solved explicitly on the detailed representation. DFN models are computationally expensive, but they play an important role in simulating fractured subsurface systems since continuum models do not fully resolve the network structure which can be a dominant factor in upscaled flow and transport properties such as breakthrough (Painter et al., 2002; Painter & Cvetkovic, 2005).

This increase in model fidelity comes at a huge computational cost because of the large number of mesh elements required to represent thousands of fractures (with sizes that can range from millimeter to kilometer) and properly resolve pressure at fracture intersections where gradients are highest. Figure 1 provides two



(a) DFN representing a field site

(b) Simple DFN with 8 fractures

Figure 1. (a) A DFN composed of around 100,000 fractures with 15 fracture families in three geological layers. (b) DFN composed of eight fractures meshed with close to 90,000 nodes. DFN = discrete fracture network.

examples showcasing the magnitude of the cost of DFN simulations. The DFN representation of a field site with layered stratigraphy shown in Figure 1a is a network made up of around 100,000 fractures drawn from 15 families with unique characteristics meshed with over 40 million nodes in a conforming Delaunay triangulation. A close up of a smaller network is shown in Figure 1b where an eight-fracture DFN is shown along with the mesh overlaying the network that contains close to 90,000 nodes. The planar representation of each fracture is discretized using 100 to 1,000 spatial nodes with optional mesh refinement at the intersections to properly resolve the high gradients in the flow field resulting from points of stagnation at intersection tips. This mesh must be further refined if in-fracture plane variability is considered because the mesh resolution needs to be sufficiently small to resolve correlations in the aperture field (Dreuzy et al., 2012; Makedonska et al., 2016). In addition to this cost, typically thousands of realizations of the DFN model must be run to bound system uncertainty stemming from variability in geophysical properties, for example, fracture network geometry and topology, as well as hydrologic properties, for example, fracture permeability.

The key aspect that sets DFN modeling apart from continuum frameworks is the recognition that fracture geometry and network topology play a critical role in determining flow and transport properties within fractured media. The mathematical construct of a graph (a set of nodes connected by edges) is one way to isolate this key aspect of DFN modeling. At the core of the DFN methodology is the conceptual model that a set of fractures, which are discrete entities, intersect one another to form a network. The mathematical construct of a graph G is another representation of this conceptual model. A graph consists of a vertex set $V(G)$ and an edge set $E(G)$, where the edge set contains pairs of vertices indicating that the two vertices in the pair are connected by an edge. There are multiple approaches for representing DFNs using graphs. However, many of these approaches seek to characterize the fracture network structure (Andresen et al., 2013; Ghaffari et al., 2011; Hope et al., 2015; Santiago et al., 2014) or simulate flow and transport on the graph itself (Cacas, 1990; Dershowitz & Fidelibus, 1999; Karra et al., 2018; Noetinger & Jarrige, 2012; Nordqvist et al., 1992).

The novelty of this paper is that we propose a DFN model reduction approach where various graph-based methods are used in conjunction with high-fidelity DFN simulations to increase the computational efficiency while retaining accuracy of key Quantities of Interest (QOI). We use four different methods—Shortest Path Pruning, Machine Learning (ML) Pruning, Physics-based Pruning, and Corrected Graph Emulator—that our team has recently developed, some of which have been previously published in this DFN model reduction framework (Hyman et al., 2017; Karra et al., 2018; Valera et al., 2018), to compare and contrast these methods in terms of accuracy and computational efficiency. Figure 2 provides an outline of the proposed DFN model reduction framework and how these five methods are utilized. We begin with a DFN (1a) and create a graph representation (1b). Then there is a junction in the proposed DFN model reduction framework. One can either query the structure of the graph, identify a backbone structure in the graph (2a), then simulate flow and transport on the corresponding DFN backbone (2b), or one can simulate flow and transport on the graph itself (3a). The DFN backbone is obtained by pruning the graph and then mapping back to the underlying fracture network. Our simulations through the approach (3a) show that the graph transport breakthrough times are consistently larger than the high-fidelity predictions and we correct this bias in the transport times as shown in (3b). In step (4), one then computes the QOI. In Figure 2, the QOI is solute transport times, that is,

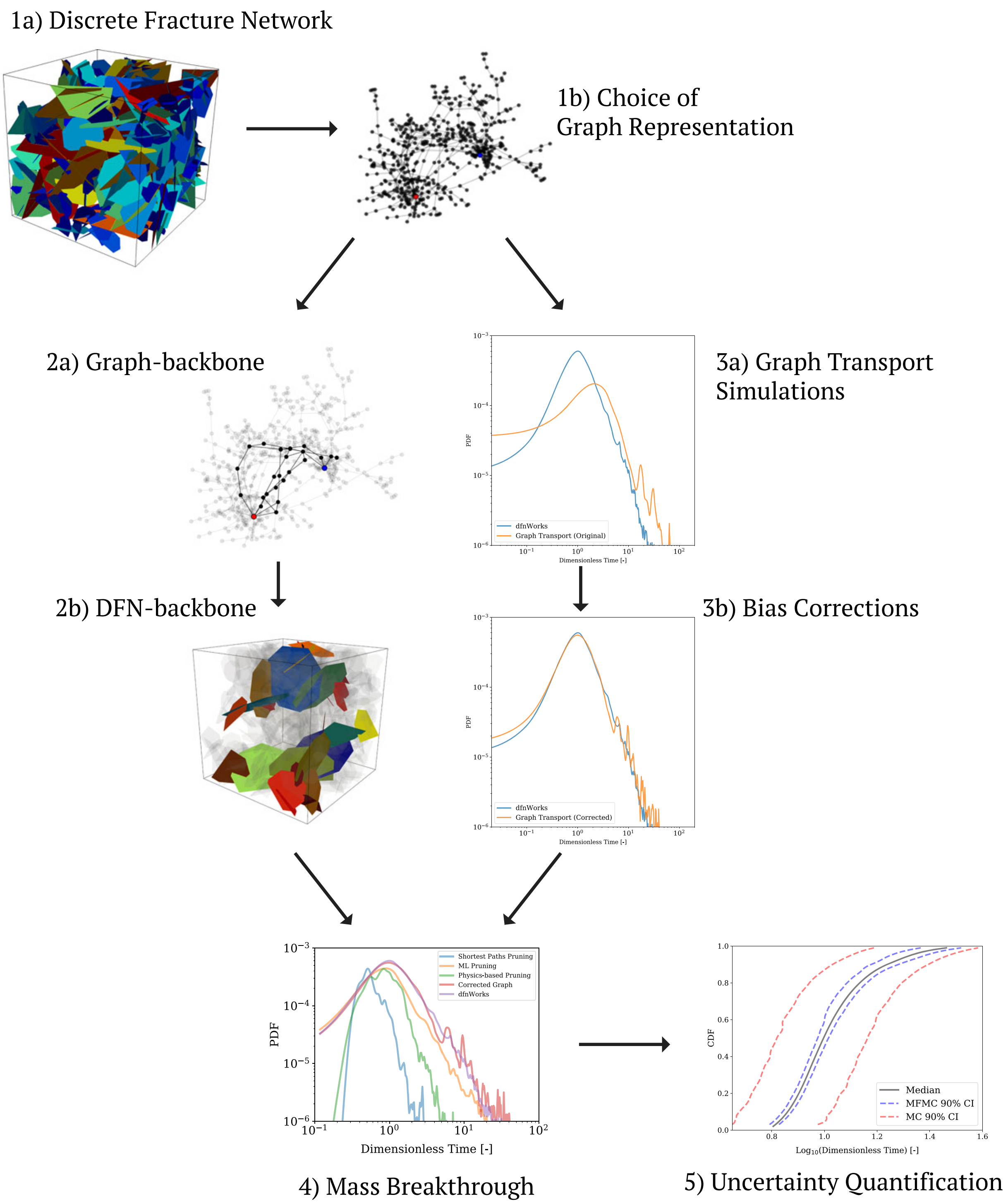


Figure 2. Schematic of proposed discrete fracture network model reduction framework. See Figures 7–9 for a closer view of the plots in steps (3b), (4), and (5), respectively.

breakthrough curves and so in (4) we have shown breakthrough curves from four methods (Shortest Path Pruning, Machine Learning (ML) Pruning, Physics-based Pruning, Corrected Graph Emulator) that we have developed and compared them to the high-fidelity DFN simulation. Our framework also utilizes a multifidelity Monte Carlo method for uncertainty quantification that uses a combination of the high-fidelity DFN and one or more of these four graph-based methods. This step is shown in (5) of Figure 2. The common thread in the methodology that sets it apart from other graph-based frameworks is that both the graph-based representation of a DFN and high-fidelity flow and transport simulations in the DFN are used and complement one another.

A key feature of the method is selecting an appropriate graph representation for the QOI. We begin by exploring two graph representations. In the first, each fracture is represented as a vertex and each fracture intersection is an edge connecting the vertices, and we discuss how it is useful for questions regarding the topological structure of the network. In the second, each intersection between fractures is represented by a vertex and fractures are represented by a collection of edges. Hyman et al. (2018) recently provided a rigorous mathematical exposition of these graph representations and showed that they are projections of a more general bipartite graph representation. Then we demonstrate the utility of the proposed method by using these graph representations to efficiently reproduce various features of the breakthrough curve, for example, first arrival, peak arrival, and scaling behavior in the tail. For problems such as hydraulic fracturing or environmental remediation, the entire breakthrough curve is typically of interest in the measurement of production or remediation efficiency. For nuclear nonproliferation problems, often it is only of interest to detect early mass arrival of gas seepage from the subsurface. We demonstrate that through the appropriate choice of mapping and graph-based techniques, we can parameterize our reduced order models to accurately predict these specific QOI with orders of magnitude higher computational efficiency than the high-fidelity DFN modeling approach. This makes our approach ideal for uncertainty quantification methods.

The rest of the paper is structured as follows. Sections 2 and 3 are based on the two types of DFN to graph mapping. The four graph-based methods that are either on the (2a, 2b) path or the (3a, 3b) path in Figure 2, use one of these two mappings and are discussed in these two sections. Section 4 compares these four methods against the high-fidelity DFN predictions in terms of accuracy and computational efficiency. A multifidelity Monte Carlo method for uncertainty quantification that builds on the high-fidelity DFN and these four graph-based methods is described in section 5. Conclusions are drawn in section 6.

2. Topological Inquiries

We begin with a graph representation that captures the connectivity (topology) of the fracture network, where fractures in the DFN correspond to a vertex in $V(G)$, intersections between fractures correspond to an edge in $E(G)$, and the rule determining the end points of the edges is that the fractures corresponding to the vertices at the edge endpoints intersect in the network. A mathematical formulation of this mapping between a fracture network F and a graph G is the following. Let $F = \{f_i\}$ for $i = 1, \dots, n$ denote a fracture network composed of n fractures. We define a mapping, ϕ, that transforms F into a graph $G(V, E)$ composed of $n = |V|$ vertices, and $m = |E|$ edges. For every $f_i \in F$, there is a unique vertex $u_i \in V$,

$$\phi : f_i \to u_i . \tag{1}$$

If two fractures f_i and f_j intersect, $f_i \cap f_j \neq \emptyset$, then there is an edge in E connecting the corresponding vertices,

$$\phi : f_i \cap f_j \neq \emptyset \to e_{ij} = (u_i, u_j) , \tag{2}$$

where $(u, v) \in E$ denotes an edge between vertices u and v. One can also include source s and target t vertices into G to incorporate flow direction. Every fracture that intersects the inlet plane $\mathbf{x}_0$ is connected to the source vertex,

$$\phi : f_i \cap \mathbf{x}_0 \neq \emptyset \to e_{si} = (s, u_i) , \tag{3}$$

and every fracture that intersects the outlet plane $\mathbf{x}_L$ is connected to the target vertex t,

$$\phi : f_i \cap \mathbf{x}_L \neq \emptyset \to e_{it} = (u_i, t) . \tag{4}$$

This choice to include boundary conditions via source and target nodes is essential if flow-based QOI, as opposed to geometric or topological properties, are considered (Neuman, 2005).

Figure 3 shows an example of this mapping for the same eight-fracture network shown in Figure 1b. Vertex colors here correspond to fracture colors in the DFN. The degree of each vertex (the number of incident edges) in the graph indicates the number of intersections with other fractures in the DFN. Topological questions, such as what is the fewest number of fractures a particle would have to pass through to get from one location to another in the system, can be easily queried on this graph representation of the DFN.

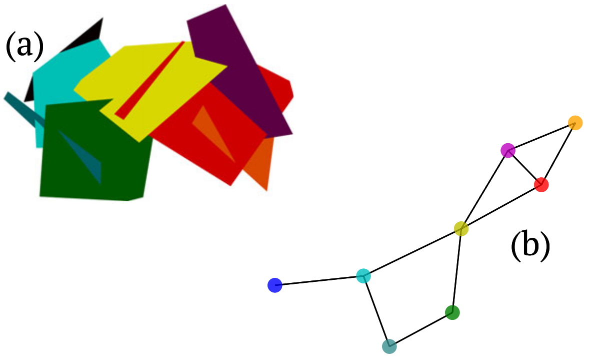


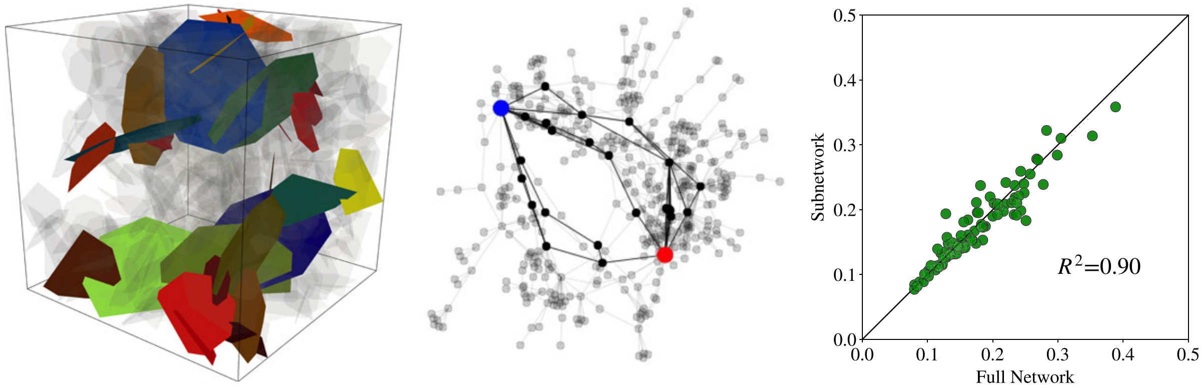
Figure 3. Topologically based graph representation of the eight-fracture network shown in Figure 1. Vertices correspond to fractures and an edge between two vertices indicates that the corresponding fractures intersect in the d.

2.1. Topological Characterization

Using this mapping, or one that is similar, topological properties of the fracture network are inherited by the graph as measurable properties such as vertex degree, efficiency, clustering, betweenness, and community structure (Andresen et al., 2013). Various studies have used this mapping to characterize and study different aspects of fracture network structure in two-dimensional (Andresen et al., 2013; Ghaffari et al., 2011; Santiago et al., 2014) and three-dimensional (Aldrich et al., 2017; Hope et al., 2015; Hyman et al., 2017; Valera et al., 2018) networks (both synthetic and natural). Ghaffari et al. (2011) linked the sensitivity of flow properties in two-dimensional networks, for example, flow field profiles, to the network structures. Andresen et al. (2013) used this mapping for quantitative comparisons between real fracture networks and models generating synthetic networks. Hope et al. (2015) showed how topological properties of the network, such as clustering (one measure of local connectivity), are related to the growth mechanism by comparing different methods for stochastic network generation. Sævik and Nixon (2017) included topological properties of two-dimensional fracture networks into analytic expressions for permeability to account for network structure. Hyman and Jiménez-Marttínez (2018) used this graph representation to aid in the investigation of the relative impact of connectivity, geometric, and hydraulic heterogeneity on transport processes. Santiago et al., (2013, 2014, 2016) proposed a method of topological analysis that measured centrality properties of nodes in the graph, which describes characteristics such as the number of shortest paths through a given node. Berrone et al. (2015) used this graph representation to partition a DFN simulations for parallelized simulations to aid in load balancing and minimize communications between processors.

2.2. Backbone Identification

Beyond characterizing topological properties of fracture networks, these graphs can be used to isolate the primary flow channels, also referred to as backbones, in the networks where the majority of flow and transport occur. In this purely topological representation of a DFN, edges weights are uniform and do not depend on in-plane geometry or hydrological properties, the shortest paths (the fewest number of edges) from source and target nodes correspond to the fewest number of fractures between inflow to outflow boundaries in the DFN. Hyman et al. (2017) used this concept to isolate subgraphs corresponding to the union of the *k* shortest paths as possible backbones in the DFN. For verification of the method, they simulated flow and transport on the subnetworks that corresponded to these subgraphs and compared it to transport through the original network. The choice to return to simulating flow and transport in the DFN, rather than on a graph representation is due to the inherent errors in simulating higher-dimensional systems with lower-dimensional representations (discussed in detail in the next section). Figure 4 provides a summary of the method for the ensemble of networks presented in section 4. Figure 4a shows a DFN and the identified backbone. Semitransparent fractures in the network are those in the compliment of the backbone. Figure 4b shows the derived graph of the network in (a) and the subgraph corresponding to the 20 shortest paths from the inflow boundary (red node) to outflow boundary (blue node), where semitransparent nodes in the graph are those in the complement of the backbone. Figure 4c presents a comparison of estimates for first particle arrival time through a sample of 100 subnetworks (20 shortest paths) and their associated original network. The high R^2 value (0.90) indicates that the two are in good agreement. These subnetworks contain significantly fewer fractures than the original network, which results in fewer degrees of freedom in the linear system of pressure and faster solver convergence. Thus, the required time for meshing, flow, and transport simulations is drastically reduced. In terms of total run time, simulating flow and transport through the *k* shortest paths subnetworks is over an order of magnitude faster than running flow and transport on the original network, but these networks still provide accurate estimates of first arrival times. For an application such as nuclear nonproliferation, first arrival times are the main QOI since one needs to determine time of arrival of gases to the surface from an underground test to determine if a test is nuclear or chemical. First arrival times are typically high in concentration leading to the best chance of detecting an isotopic signature expediently. Since the fracture network characteristics of a site will only be known statistically, fast simulations that can predict first arrival times accounting for and exploiting the topology of the fracture network are needed. Identifying paths of least resistance,



(a) Subnetwork (b) Subgraph (c) First Breakthrough

Figure 4. (a) Subnetwork and (b) subgraph corresponding to the 20 shortest paths from the inflow boundary (red node) to outflow boundary (blue node). Semitransparent fractures in the network and nodes in the graph are removed. (c) A comparison of estimates for first particle arrival time through a sample of 100 subnetworks and their associated original network shows they are in good agreement. Meshing, flow, and transport simulation times on these subnetwork are over an order of magnitude less than that required for the original network. Figure modified from (Hyman et al., 2017).

for instance using graph theory, and linking them to the fastest transport has also been performed in two and three-dimensional heterogeneous porous media (Rizzo & de Barros, 2017; Tyukhova & Willmann, 2016).

While the method of Hyman et al. (2017) is useful for finding backbones where the fastest travel occurs, it does not do well in identifying backbones that carry the majority of the flow. One way to resolve this problem is including flow and transport information into the graph as edge weights. Aldrich et al. (2017) used this mapping to identify backbones in fracture networks by incorporating information from particle transport simulations. By setting the edge weights to be inversely proportional to the amount of mass (number of particles) that passed between the two fractures, they constructed a weighted-directed graph that represented the flow structure. Figure 5 shows an example of such a network, which is commonly called a flow topology graph (FTG). Arrows on edges show the direction in the graph and indicate the direction of flow from one fracture to another. Various metrics can be implemented on the FTG to find the weighted-shortest paths between the inflow and outflow boundaries (blue and red vertices, respectively) and is highlighted here by the set of green edges. The fractures which correspond to the vertices along this path are a backbone where a majority of transport occurs. This method for backbone identification, however, is computationally expensive because flow, transport, and postprocessing of particle trajectories have to be performed.

To overcome this hindrance in the application of the method, Valera et al. (2018) trained machine learning classifiers using these backbones to create a more efficient workflow. Specifically, they trained two supervised machine learning algorithms, support vector machines, and random forest, to classify individual fractures, here represented as vertices in the graph, as being members of the backbone. The features used for classification were a mix of global topological properties (betweenness centrality; source-to-target current flow; and source-to-target simple paths), local topological properties (degree centrality), geometric (projected volume), and hydrological attributes (permeability). In contrast to the method of Aldrich et al. (2017), which requires flow and transport to be performed and can take several hours to run on a single network, once the classifiers are trained (in a matter of minutes), the supervised classification can then be used to identify potential backbones in a matter of seconds. The quality of classification can be measured using two metrics, precision [*True positives / (True positives + False Positives)*] and recall [*True positives / (True positives + False Negatives)*]. In general, there is a trade-off between precision and recall, higher precision results in lower recall and vice versa,

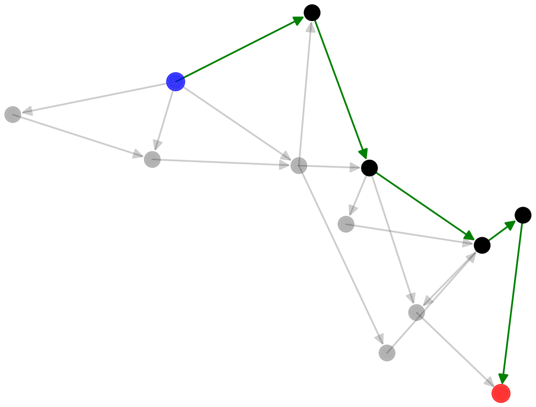


Figure 5. Path analysis on a flow topology graph (FTG). Flow and transport on a discreet fracture network is represented by a weighted-directed graph where the weights are inversely proportional to the amount of mass transport passing from one fracture to another. Various metrics can be implemented on the FTG to find the weighted-shortest paths between the inflow and outflow boundaries (blue and red vertices respectively) and it is highlighted here by the set of green edges. The fractures that correspond to the vertices along this path are the backbone where a majority of transport passes through the system.

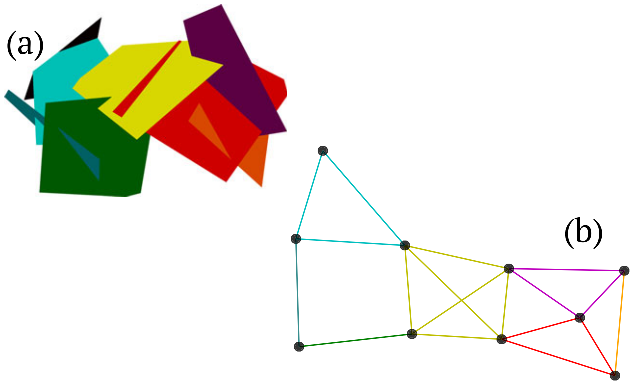


Figure 6. Hydrological-based graph representation of the eight-fracture network shown in Figure 1. Black vertices correspond to intersections and an edge between two vertices indicates that the corresponding fractures intersect in the discrete fracture network (DFN). Edge colors correspond to fracture colors in the DFN.

that can be controlled by adjusting the methods' parameters. One can reduce the number of fractures in the backbone with either method, (high precision and low recall) but this resulted in poor predictions of travel time distributions. On the other hand, fewer fractures can be removed by including more fractures into the backbone, (low precision and high recall) but the predictions of travel time distributions are better. A useful feature of random forest is that it returns a measure of feature importance, which is the ratio between the number of samples routed to a decision node involving that feature in any of the trees of the ensemble over the total number of samples in the training set. For backbone classification the most important features were found to be topological (betweenness centrality, source-to-target current flow, source-to-target simple paths, and degree centrality) rather than the geometric or hydrological properties (projected volume and permeability).

3. Hydrological Modeling

In this section, we explore a different mapping between DFN and graph and discuss how it is useful for answering hydrological questions as opposed to purely topological ones. In this mapping, intersections in the DFN correspond to nodes in the graph and fractures are represented as a clique of edges. These traits allow for geometric and hydrological properties to be inherited by the graph edges. Below we provide the mathematical formulation of this hydrologically based mapping between the fracture networks F and a G. Let $F = \{f_i\}$ for $i = 1, \dots, n$ denote a DFN composed of n fractures. We define a mapping, ψ, that transforms F into a graph $G(V, E)$ composed of $N = |V|$ nodes, and $M = |E|$ edges. If two fractures f_i and f_j intersect, $f_i \cap f_j \neq \emptyset$, then there is a node $u \in V$,

$$\psi : f_i \cap f_j \to u . \tag{5}$$

that represents the line of intersection. Similarly, if f_i intersects the inflow or outflow plane, then a node is added to V that represents the line of intersection. If $f_i \cap f_j \neq \emptyset$ and $f_i \cap f_k \neq \emptyset$, then there is an edge in E connecting the corresponding nodes,

$$\psi : f_i \cap f_j \neq \emptyset \text{ and } f_i \cap f_k \neq \emptyset \to e(u, v), \tag{6}$$

where $e(u, v) \in E$ denotes an edge between nodes u and v. Under this mapping, each fracture in the network is represented by a clique in G. Figure 6 shows an example of this mapping for the same eight-fracture network shown in Figure 1b. Edge colors here correspond to fracture colors in the DFN and black vertices represent fracture intersections.

3.1. Reduced Order Models of DFNs: Graph Flow and Transport Predictions

One important distinction between the mapping ϕ (equations (1)–(4)) and ψ is that flow and transport can be simulated on the graph created via ψ. In fact, this particular mapping is similar to the one used in conventional pipe-flow networks (Dershowitz & Fidelibus, 1999; Noetinger & Jarrige, 2012; Nordqvist et al., 1992). Simulating transport in hydrologic systems is critical for numerous applications including groundwater contamination (Cvetkovic et al., 2004; National Research Council, 1996; Neuman, 2005; O'Malley & Vesselinov, 2014a), hydrocarbon extraction (Hyman, Jiménez-Martínez, et al., 2016; Karra et al., 2015), and detection of underground nuclear explosions (Jordan et al., 2014). These applications often demand the use of large-scale models making the simulation of transport computationally demanding (Lichtner et al., 2015), so the use of computationally efficient graph-based models can provide significant benefits to techniques that require a large number of model runs such as model calibration (Lin et al., 2017) or uncertainty quantification (O'Malley & Vesselinov, 2014b).

Simulating flow and transport on the graph is a two-step process (Karra et al., 2018). The first step requires solving a flow equation on the graph that produces something analogous to the velocity field in a standard (nongraph-based) computational hydrology approach. The flux from vertex u to vertex v is $Q_{uv} = w_{u,v}(P_u - P_v)$ where P_u and P_v are the pressures at vertices u and v, respectively, and $w_{u,v}$ is a coefficient that describes how readily the fluid can move from the intersection associated with vertex u to the intersection associated with vertex v. The edge weight $w_{u,v}$ is computed using geometric (length and aperture) and hydrological (permeability) properties of the DFN; details of various methods as to how these attributes can be combined can be found in (Dershowitz & Fidelibus, 1999; Karra et al., 2018; Noetinger & Jarrige, 2012; Nordqvist et al., 1992). Conservation of mass is enforced by summing the fluxes at a vertex and setting the sum equal to zero, resulting in a set of linear equations for the pressures, which can be readily solved using standard techniques from numerical linear algebra.

The second step is simulating transport using particle tracking. Particles move from one vertex to a randomly chosen neighboring vertex that has a lower pressure in a flux-weighted fashion (i.e., the probability of going from vertex *u* to vertex *v* is proportional to the flux from *u* to *v*) and complete mixing is adopted as opposed to streamline routing (Kang et al., 2015; Park et al., 2003; Stockman et al., 1997). The travel time from one vertex to another is computed assuming a straight line path between the centers of the two corresponding intersections with a velocity that is proportional to the pressure difference. Each particle moves from a vertex associated with the inflow boundary through a series of downgradient vertices until reaching a vertex associated with the outflow boundary. During this process, the time to travel between vertices is accumulated to finally produce a breakthrough time.

3.2. ROM Discrepancies

The graph-based model of transport provides a highly efficient means of simulating transport through fractured hydrologic systems. Karra et al. (2018) compared simulation times in a variety of fracture networks and observed speed up of up to 5 orders of magnitude. While the graph-based model is much faster than the DFN model, it comes with a compromise in the accuracy of the breakthrough times. This graph representation is a coarse representation of possible flow paths through the network, which can introduce biases in the transport times from one fracture intersection to another. In particular, the simplified representation of in-plane fracture geometry constrains the path that particle can take through the graph. In the full DFN, there are many paths a particle can take from one intersection to another (each with variable velocity along the path), whereas in the graph there are far fewer paths (each with constant velocity). Another discrepancy introduced by the graph representation is that intersections, which are lines in the high-fidelity model, are represented as single vertices in the graph. In the DFN, pressure can vary along an intersection, while in the graph, it cannot. In turn, the distribution of velocities in the graph is much smaller than that in the DFN. An interesting limitation of the graph model is linked to the construction of the graph. In the graph, flow on dead-end fractures (fractures that only connect to one other fracture) is completely omitted by construction; there are no edges in the graph that correspond to dead-end fractures. In two-dimensional fracture networks, these dead-end fractures are no-flow regions and removing them does not influence flow and transport properties in the absence of diffusion. However, in three-dimensional networks, there is flow in dead-end fractures due to local recirculation, which are the result of pressure gradients along fracture intersections. These recirculation zones can influence the tails of the advective transport time distributions and are particularly important if diffusion is also considered. Finally, if in-fracture aperture variability is considered in the DFN model, it is not clear how to properly upscale the variability into a single edge weight.

In small systems, the discrepancies between transport in the DFN and graph resulting from these issues appear to be small. However, as the system size increases, so does the difference between the two models (Karra et al., 2018). Nonetheless, these graph representations remain useful and are particularly powerful when combined with high-fidelity simulations. Even though the graph-based model produces biased predictions for large DFNs, these predictions can still be useful because they are highly correlated with the predictions of the DFN model. That is, the two models have different means, but their variations around those means are similar. This interesting property allows for the development of multifidelity techniques that combine DFN and graph-based flow and transport simulations to yield accurate predictions of QOI with massive reductions in computational costs. In this subsection, we describe a few recently developed methods for doing the latter. A variety of alternative methods (e.g., Gaussian Process Regression, Quiñonero-Candela & Rasmussen, 2005) could also be used for model reduction that may have different advantages depending on the context.

3.2.1. Physics-Based System Reduction

Even though the pressures on the graph are not identical to the high-fidelity DFN, they do exhibit the same overall structure; the direction of flow and relative fluxes are consistent between the two models. This similarity lends itself to another approach for system reduction of the DFN (backbone identification), one that is the next logical step from the methods described in section 2, which used only topological information known a priori. In this approach, one solves the flow equation only on the graph to perform the system reduction.

The flow solution on the graph (knowing the pressure at every vertex) defines a flux for each edge. Thus, one can identify subnetworks with high flux values, where large amounts of mass transport should occur. This yields a smaller, pruned network, and instead of then using the full network, this pruned network is used for high-fidelity simulations. Hence, there is computational gain in each component of the simulation, viz, network generation, meshing, flow simulation, and transport calculation. The smaller the size of the pruned network, more is the gain. A straightforward way to find these subnetworks is applying a threshold parameter ε to a graph G and identifying the maximal subgraph (A maximal subgraph is a subgraph where there does not exist a larger subgraph containing this subgraph) $G'(V', E')$ such that all of its edges $e(u, v)$ have a flux $q(u, v)$ greater than the threshold,

$$\text{if } q(u,v) > \varepsilon \text{ then } e(u,v) \in E', \tag{7}$$

and

$$V' = \{u\} \text{ such that } u \in E', \tag{8}$$

Then, as described in section 2, one can simulate flow and transport in the DFN that corresponds to G'. We refer to this method as *physics-based system reduction*, because it takes physical observables into account in the backbone identification. A similar method was proposed by Maillot et al. (2016), but they considered flow on a high-fidelity DFN rather than on a graph representation.

The choice of ε determines how aggressive is the system reduction strategy and also influences the computational savings, that is, reduction in network size. One can adjust the value of ε so the discrepancies in the breakthrough curves at a given percentile is within a specified tolerance, but still removing as much of the network as possible. The most aggressive way to apply this methodology is to increase ε until the source and target are disconnected, which is the critical flux value such that there is a connected path through the graph such that all flux values are greater than this value. The fractures along this path would constitute another type of backbone where the fastest transport occurs. However, one does not need to raise ε to this value. Smaller values can be adopted and thus retain more of the network, and in turn retain better agreement between a quantity of interest in the original network and the induced subnetwork. This particular method is useful for cases where the full breakthrough curve is of interest, such as contaminant remediation or hydrocarbon extraction, instead of just the first arrival time of the plume.

3.2.2. Training Graph Transport

Each high-fidelity DFN simulation contains a wealth of information about transport times from one fracture intersection to another and corrections at this level can be used to improve the graph transport model. Karra et al. (2018) capitalized on this information and used detailed Lagrangian information to correct for the bias in the graph transport. They considered an ensemble of fracture networks with approximately 500 fractures each and obtained detailed information about the trajectories in one member of this ensemble. They extracted information about the fracture intersections that each particle passed through and how long it took to travel from one intersection to another for over 1,000 particles in a single realization. Then, the travel time from one intersection to another was used to improve the prediction of the same quantity in the graph transport model. For each pair of intersections that a particle in the DFN model traveled through, a time t_D was obtained and compared to the travel time predicted by the graph model t_G. They found that a power law relationship could be used to improve the graph transport model prediction, t_G, so that it more closely resembles t_D,

$$t_D \approx C t_D^{\alpha} \tag{9}$$

where C has units of fractional time ($t^{1-\alpha}$) and α is the exponent that is estimated from the data. The exponent is estimated by using linear regression of the quantities log t_D and log t_G.

Figure 7 shows the result of training on one high-fidelity DFN run, obtained from the ensemble described in the next section. Time has been made dimensionless by dividing the breakthrough times by the median mass breakthrough times in the DFN, colored blue. Note that while the pure graph transport model (orange) has a significant bias, the version of the graph transport model that has been trained on a single DFN realization (green) is able to predict the cumulative breakthrough curve much more accurately. The realization of the DFN that was used to generate Figure 7 differs from the realization that was used to perform the training process.

4. Method Comparison

All of the previously described graph-based methods have unique advantages and disadvantages. In this section, we compare and contrast them in the context of one of the primary quantities of interest in subsurface

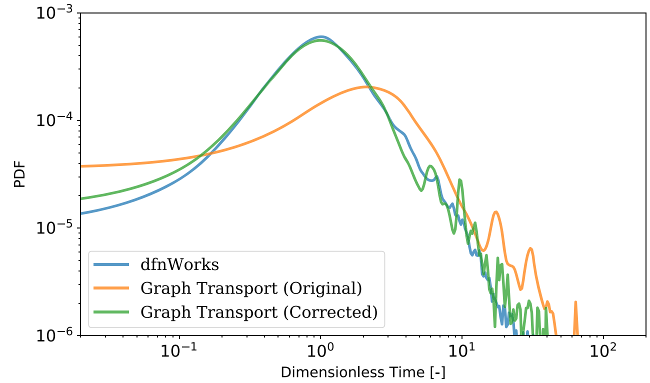


Figure 7. An example of the cumulative breakthrough curves predicted by the DFN model (blue), the graph transport model (orange), and the graph transport model that has been trained on one DFN realization (green). Note that training on a single DFN realization substantially improves the accuracy of the graph transport model's predictions on the other members (such as this representative example shown in green) of the ensemble.

flow and transport modeling, namely, solute transport times. For the comparison, we construct a set of generic networks. We use the computational suite dfnWorks (Hyman, Karra, et al., 2015) to generate the fracture networks, solve the steady state flow equations, and determine transport properties through the network. dfnWorks uses the feature rejection algorithm for meshing (FRAM; Hyman et al., 2014) to generate three-dimensional fracture networks and the LaGriT meshing toolbox (LaGriT, 2013) to generate conforming Delaunay triangulation of the DFN. The parallelized subsurface flow and reactive transport code PFLOTRAN (Lichtner et al., 2015) is used to numerically integrate the steady state governing flow equations. An extension of the WALKABOUT particle-tracking method (Makedonska et al., 2015; Painter et al., 2012) is used to determine pathlines through the DFN and simulate solute transport. Details of the suite, its abilities, applications, and references for detailed implementation are provided in Hyman, Karra, et al. (2015). The networks are composed of circular fractures with uniformly random orientations. The DFN is in a cube of side length 15 m. The fracture radii r [m] are sampled from a truncated power law distribution with exponent $1 + \alpha$ and upper and lower cutoffs $(r_u; r_0)$. We choose $\alpha = 2.6$, a lower cut off $r_0 = 1$ and the upper cut off $r_u = 5$. Fracture centers are sampled uniformly throughout the domain. There are 499 fractures in the network. The networks are sparse, with a P_{32} value (fracture surface area over total volume) of around 2. This ensures that networks are dense enough to have multiple paths connecting the inflow and outflow boundaries. Fracture apertures are constant for each fracture, but positively correlated with their radii through a power law relationship, $b = \gamma r^{\beta}$, where $\gamma = 5.0 \times 10^{-5}$ and $\beta = 0.5$ are dimensionless parameters (Bogdanov et al., 2007; de Dreuzy et al., 2002; Hyman, Aldrich, et al., 2016; Hyman, Painter, et al., 2015; Joyce et al., 2014; Wellman et al., 2009).

Within each fracture, we adopt the cubic law to relate the permeability of each fracture to its aperture. Flow is driven through the network by applying a pressure difference of 1 MPa across the domain aligned with the x axis. No-flow boundary conditions are applied along lateral boundaries, and gravity is not included in these simulations. These boundary conditions along with mass conservation and Darcy's law are used to form an elliptic partial differential equation for steady state distribution of pressure within each network

$$\nabla \cdot (k \nabla P) = 0. \tag{10}$$

Once the distribution of pressure and volumetric flow rates are determined by numerically integrating (10), the methodology of Makedonska et al. (2015) and Painter et al. (2012) is used to determine the Eulerian velocity field $\mathbf{u}(\mathbf{x})$ at every node in the conforming Delaunay triangulation throughout each network.

The spreading of a nonreactive conservative solute transported is represented by a cloud of passive tracer particles, that is, using a Lagrangian approach. The imposed pressure gradient is aligned with the x axis, and thus, the primary direction of flow is in the x direction. Particles are released from locations in the inlet plane $\mathbf{x}_0$ at time $t = 0$ and are followed until they exit the domain at the outlet plane $\mathbf{x}_L$. The trajectory $\mathbf{x}(t; \mathbf{a})$ of a particle starting at $\mathbf{a}$ on $\mathbf{x}_0$ is given by the advection equation, $\mathrm{d}\mathbf{x}(t; \mathbf{a})/\mathrm{d}t = \mathbf{v}(t; \mathbf{a})$ with $\mathbf{x}(0; \mathbf{a}) = \mathbf{a}$ where the Lagrangian velocity $\mathbf{v}(t; \mathbf{a})$ is given in terms of the Eulerian velocity $\mathbf{u}(\mathbf{a})$ as $\mathbf{v}(t; a) = \mathbf{u}$. The mass represented by each particle $m(\mathbf{a})$ and the breakthrough time at the outlet plane, $\tau(\mathbf{x}_L; \mathbf{a})$ of a particle that has crossed the outlet plane, $\mathbf{x}_L = (L, y, z)$ can be combined to compute the total solute mass flux $F(t)$ that has broken through at a time t,

$$F(t, \mathbf{x}_L) = \frac{1}{M} \int_{\Omega_a} \mathrm{d}m(\mathbf{a}) \delta[t - \tau(\mathbf{x}_L, \mathbf{a})], \tag{11}$$

where Ω_a is the set of all particles, M is the total mass put into the system, $M = \sum_{\Omega} m(\mathbf{a})$, and δ is the Dirac delta function. Mass is distributed uniformly among the particles. Network generation, meshing, flow, and transport simulations takes around 30 min using 16 cores per DFN.

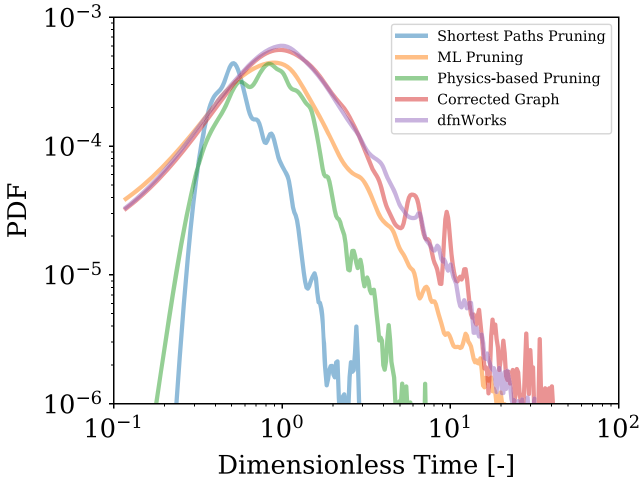


Figure 8. Comparison of breakthrough curves in backbones identified using the shortest paths (section 2.2), machine learning based on the flow topology graph (section 2.2), physics-based system reduction (section 3.2.1), corrected graph transport (section 3.2.2) on a graph representation of the full network and the full discrete fracture network.

For the comparison we consider the algorithm to find shortest paths (section 2.2), machine learning based on the flow topology graph (section 2.2), physics-based system reduction (section 3.2.1), and corrected graph transport (section 3.2.2) applied to one instance in the set of networks. Figure 8 shows the breakthrough curves for each method; Shortest paths (blue), Machine learning (yellow), Physics-based (green), Corrected graph (red), and the original DFN (purple). Information about the network reduction is provided in Table 1. The plots of $F(t, \mathbf{x}_L)$ are generated using a kernel density transformation of the raw data (Silverman, 1982). Time has been nondimensionalized by the peak breakthrough time of F in the original DFN, $t = t/t_{\text{peak}}$ where $t_{\text{peak}} = \underset{t}{\operatorname{argmax}}[F(t, \mathbf{x}_L)]$.

Each of the curves displays different behavior relative to the original breakthrough curve. The subnetwork corresponding to the shortest paths (20 shortest paths in this case) from source to target in the topological mapping match the first arrival time but quickly deviate. The spread of particles in this plume is also very narrow compared to the original network. Recall that this is what the shortest paths algorithm was designed to do —isolate the subnetwork where the fastest transport occurs. The subnetwork identified by this method contains 27 fractures (≈5%) of the original network and takes around 3 min for meshing, flow and transport simulations, which is a factor of 10 speedup. Identification of the backbone takes less than a second. The machine learning method (support vector machine) was trained on the flow topology graphs of a subset of the networks, and details of the selected method parameters for the support vector machine are provided in Valera et al. (2018). The predicted breakthrough matches well at early times, but then deviates. However, transport through the machine learning identified network provides a good estimate of the peak arrival time but under predicts the peak dosage. Recall that the training data for the machine learning algorithm seeks to find where the majority of flow goes. Flow and transport simulations on this network took around 7 min and yielded speedup of a factor of 4. However, note that this speed up is biased because the construction of the training set requires that flow and transport simulations be performed on a set of complete networks. Next, we apply the physics-based pruning method to find a backbone. Here the backbone corresponds to a subgraph identified using the critical flux or critical threshold value as discussed above. Simulations on this subnetwork took around 5 min, a speedup of a factor of 6. Transport through this subnetwork matches the earliest times well and a decent estimate for peak breakthrough time but underestimates the peak dosage. Recall that this backbone corresponds to fast paths in graph so the matching of the early times is expected. However, the distribution is much narrower than the original network and the machine learning network, indicating a more uniform velocity field within the network. Finally, breakthrough times in the corrected graph are in excellent agreement with the times observed in the original network matching early, peak, and even tailing behavior. Simulation on the graph takes a matter of seconds, a speedup of 3 orders of magnitude resulting from not having to mesh or perform flow and transport on the DFN. Similar to the machine learning backbone, this speedup time is biased because meshing, flow, and transport have to be performed on at least one sample from the set of original fracture networks to determine the correction factor in equation (9). However, since the premise of our

Table 1
Comparison of Subnetwork Properties and Approximate Speedup

Method	Number of fractures	% of original network	Speedup factor
Shortest paths	27	5	10
Machine learning	145	29	4*
Physics-based pruning	167	33	6
Corrected graph	499	100	10^3*
Original DFN	499	100	1

Note. Asterisk * indicates that at least one training run on the original DFN is required.

work is that we want to fit our algorithms in a UQ framework where thousands of realizations are needed, the time for this training step is insignificant when compared to the time taken to perform the flow and transport calculations on all the realizations. For this reason, we have not included this time in the speedup comparison.

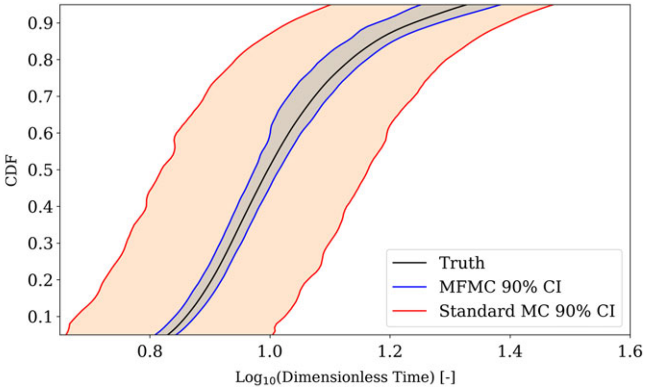


Figure 9. Breakthrough results from the Multifidelity Monte Carlo (MFMC) method using high-fidelity discrete fracture network (DFN) and graph transport emulator compared against standard Monte Carlo method. Ninety percent confidence intervals are shown in both cases. Truth in this case is based on large number of DFN realizations.

5. Multifidelity Monte Carlo for Uncertainty Quantification

In order to keep the computational cost low while reducing uncertainty in predicting the QOI, we use a multifidelity Monte Carlo method built on O'Malley et al. (2018) as the last step (step 5 of Figure 2) of our DFN model reduction framework. Multifidelity Monte Carlo methods (Peherstorfer et al., 2016) aim to exploit models of varying fidelity to estimate a statistical quantity of interest (such as the expected first or median breakthrough time in the context of a hydrologic transport model). Since lower-fidelity models are often much faster than high-fidelity models, there is a trade-off in the Monte Carlo estimator between performing a small number of costly-but-accurate model runs or a large number of inexpensive-but-biased model runs. These methods can be used to provide an optimal balance for this trade-off in the sense that for a fixed computational cost, the variance of the multifidelity Monte Carlo estimator is minimized by appropriately choosing the number of high and low-fidelity model runs. It is also important to note that the multifidelity Monte Carlo estimator is unbiased even if the lower-fidelity models are biased. The lower-fidelity models must have a good combination of speed and correlation with the high-fidelity model in order for them to be effective within this framework.

O'Malley et al. (2018) demonstrated that the graph flow and transport models meet this requirement and therefore can be used as part of a multifidelity Monte Carlo method to produce unbiased estimates of quantities such as median breakthrough time. In this context, the multifidelity approach essentially utilizes the DFN model to produce an unbiased estimator of the quantity of interest and then the graph-based model is used to perform a large number of model runs. The combined effect is that for a fixed computational cost, the multifidelity Monte Carlo estimator is unbiased (like standard Monte Carlo), but has 2 orders of magnitude lower variance. It is important to note that even for a very small number of high-fidelity DFN model runs that are not fully representative of the ensemble, the multifidelity Monte Carlo estimator is unbiased. The unbiasedness is a mathematical property of the estimator that does not require the high-fidelity model runs to fully explore the parameter space (one realization of the high-fidelity model is enough to ensure the estimator is unbiased). Figure 9 shows a comparison between the multifidelity Monte Carlo method and standard Monte Carlo to estimate the breakthrough curve for particles advecting through a fracture network. Note that the confidence intervals in this plot are confidence intervals for the two Monte Carlo estimators, so being tighter around the truth is better. The multifidelity Monte Carlo method samples from both high-fidelity DFN and the graph transport emulator discussed in section 3. The standard Monte Carlo refers to sampling realizations of the DFN and using only the high-fidelity model to estimate the quantity of interest. The multifidelity estimator has a tighter confidence interval around the truth because it could explore (and average out) more variability in the system by using nine DFN model runs and 10,000 graph-based model runs. This is in contrast to the standard Monte Carlo estimator which, for the same computational cost, could only explore 10 DFN model runs, leaving it vulnerable to outliers and creating a large confidence interval. The contrast in confidence intervals for the two estimators indicates that the multifidelity Monte Carlo method reduces the uncertainty in the estimate despite the fact that both approaches had the same computational cost. The multifidelity Monte Carlo approach that we use is not restricted to the graph transport emulator as the low-fidelity model but can be used with any of the other methods discussed in sections 2 and 3. Additionally, this approach is not limited to just two models with differing fidelities but can be used for a combination of multiple high- and low-fidelity models.

6. Conclusion

Modeling flow and transport through fractured systems is a challenge because representing thousands of fractures and their intersections requires computationally intensive meshing, flow, and transport algorithms.

Fractures are difficult to properly include in continuum formulations since they are often the primary flow pathways and the topology and hydrological properties of a fracture network often dominate system behavior but are smoothed out by the continuum assumption.

Graphs provide a simple and elegant way to characterize, query, and interrogate fracture network connectivity. When used in a DFN model reduction framework with high-fidelity DFN, they provide a way to characterize, query, and efficiently simulate flow and transport through fractured media. We proposed an approach where various graph representations are used in conjunction with high-fidelity DFN simulations to increase computational efficiency while retaining accuracy of key quantities of interest. We considered two different mappings between fracture networks and graphs and showed each of them is useful for different studies. Selection of the mappings depends on the quantities of interest, such as first arrival time of a solute plume, topological properties, or if the entire breakthrough curve needs to be considered. When only the first arrival times are of interest, a mapping that is based solely on topology can be used to determine shortest paths through a fracture network. A computationally efficient pruned DFN predicts first arrival times that are in good agreement with the original DFN and underscores that large-scale structure is the dominant factor controlling where transport goes within the network. Predictions of first arrivals are critical for applications such as nuclear nonproliferation. We also discussed that graphs can be used to increase the computational efficiency of the full DFN through graph partitioning for parallel computations. Furthermore, they can be used to identify backbones in high-fidelity simulations and topological characterization of fracture networks.

On the other hand, using a mapping that allows graph edges to inherit hydrological properties such as permeability and in-plane geometry can be used to simulate flow and transport in a computationally efficient manner. These models have at least 3 orders of magnitude computational savings for a 500-fracture network but exhibit systematic discrepancies from the full DFN. Karra et al. (2018) show that the speedup can go up to 5 orders of magnitude for larger fracture networks. We discussed a method to correct this systematic discrepancy to better match the high-fidelity DFN breakthrough curve. In most subsurface applications fracture characteristics are only known statistically and numerous realizations are needed to bound flow and transport in the system. The presented DFN model reduction framework provides an incredibly efficient workflow to bound system behavior and quantify uncertainty by combining DFN and graph-based flow and transport simulations. Furthermore, this mapping can also be used to prune the DFN using physics rather than topology alone (as was used for the early arrival time case). This physics-based pruning can be used to generate breakthrough curves that capture the behavior of the entire breakthrough curve as needed by applications such as contaminant remediation or energy extraction where both early and late arriving mass is of interest, while still reducing the size of the network considerably.

We also discussed applying a multifidelity Monte Carlo approach in our framework that combines the fast graph-based reduction models with the high-fidelity computationally intensive DFN simulations. This method is ideal for reducing the uncertainty in predictions for a fixed computational cost.

There are, however, aspects of DFN modeling that graphs are not well suited for. For example, the dimension reduction associated with graph-based flow modeling makes the inclusion of internal fracture aperture variability challenging. Recent three-dimensional DFN modeling studies have shown that these effects can influence upscaled system attributes such as effective permeability and transport times (Dreuzy et al., 2012; Makedonska et al., 2016). While there are methods to include dispersion based on fracture roughness (Nordqvist et al., 1992), they have yet to be benchmarked against fully resolved three-dimensional DFN models.

Although for single-phase saturated flow through a fracture network, the graph emulator method provides by far the greatest speedup (with at least 3 orders of magnitude) and best match to the entire breakthrough curve, for more complex physics such as multiphase flow and reactive transport, we will have to revisit this methodology. On the other hand, with the pruning-based methods, one can directly use the pruned DFN to model complex physics without any modifications to the workflow, for example, using existing subsurface simulators, while obtaining 1 to 2 orders of magnitude in speedup.

In conclusion, the ability of graphs to represent and characterize fractured systems makes them a powerful tool for many applications. When paired with three-dimensional DFN modeling, they make it possible for a much more efficient workflow that lends itself to robust uncertainty quantification and characterization of subsurface flow and transport.

Acknowledgments

The authors thank Los Alamos National Laboratory LDRD program for the support through project 20170103DR and for the support provided by the Department of Energy (DOE) Basic Energy Sciences program (DE-AC52-06NA25396). Data used in this paper are available at https://gitlab.com/satkarra/Viswanathan_etal_2018_WRR_graphs_data. S.S. gratefully acknowledges support from the Center for Nonlinear Studies at Los Alamos National Laboratory.

References

Aldrich, G., Hyman, J. D., Karra, S., Gable, C. W., Makedonska, N., Viswanathan, H., et al. (2017). Analysis and visualization of discrete fracture networks using a flow topology graph. *IEEE Transactions on Visualization and Computer Graphics*, *23*(8), 1896–1909.

Andresen, C. A., Hansen, A., Le Goc, R., Davy, P., & Hope, S. M. (2013). Topology of fracture networks. *Frontiers in Physics*, *1*, 7.

Berrone, S., Pieraccini, S., & Scialo, S. (2013). A PDE-constrained optimization formulation for discrete fracture network flows. *SIAM Journal on Scientific Computing*, *35*(2), B487–B510.

Berrone, S., Pieraccini, S., Scialò, S., & Vicini, F. (2015). A parallel solver for large scale DFN flow simulations. *SIAM Journal on Scientific Computing*, *37*(3), C285–C306.

Bogdanov, I., Mourzenko, V., Thovert, J.-F., & Adler, P. (2007). Effective permeability of fractured porous media with power-law distribution of fracture sizes. *Physical Review E*, *76*(3), 36309.

Cacas, M. C. (1990). Modeling fracture flow with a stochastic discrete fracture network: Calibration and validation: 1. The flow model. *Water Resources Research*, *26*(3), 479–489.

Cvetkovic, V., Painter, S., Outters, N., & Selroos, J. O. (2004). Stochastic simulation of radionuclide migration in discretely fractured rock near the Äspö hard rock laboratory. *Water Resources Research*, *40*, W02404. https://doi.org/10.1029/2003WR002655

de Dreuzy, J.-R., Darcel, C., Davy, P., & Bour, O. (2004). Influence of spatial correlation of fracture centers on the permeability of two-dimensional fracture networks following a power law length distribution. *Water Resources Research*, *40*, W01502. https://doi.org/10.1029/2003WR002260

de Dreuzy, J.-R., Davy, P., & Bour, O. (2001). Hydraulic properties of two-dimensional random fracture networks following a power law length distribution 2. Permeability of networks based on lognormal distribution of apertures. *Water Resources Research*, *37*(8), 2079–2095.

de Dreuzy, J.-R., Davy, P., & Bour, O. (2002). Hydraulic properties of two-dimensional random fracture networks following power law distributions of length and aperture. *Water Resources Research*, *38*(12), 1276.

Dershowitz, W., & Fidelibus, C. (1999). Derivation of equivalent pipe network analogues for three-dimensional discrete fracture networks by the boundary element method. *Water Resources Research*, *35*(9), 2685–2691.

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*, *117*, B11207. https://doi.org/10.1029/2012JB009461

Erhel, J., de Dreuzy, J.-R., & Poirriez, B. (2009). Flow simulation in three-dimensional discrete fracture networks. *SIAM Journal on Scientific Computing*, *31*(4), 2688–2705.

Ghaffari, H., Nasseri, M., & Young, R. (2011). Fluid flow complexity in fracture networks: Analysis with graph theory and LBM. arXiv preprint arXiv: 1107.4918.

Hope, S. M., Davy, P., Maillot, J., Le Goc, R., & Hansen, A. (2015). Topological impact of constrained fracture growth. *Frontiers in Physics*, *3*, 75.

Hyman, J., Aldrich, G., Viswanathan, H., Makedonska, N., & Karra, S. (2016). Fracture size and transmissivity correlations: Implications for transport simulations in sparse three-dimensional discrete fracture networks following a truncated power law distribution of fracture size. *Water Resources Research*, *52*, 6472–6489. https://doi.org/10.1002/2016WR018806

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.

Hyman, J. D., Hagberg, A., Srinivasan, G., Mohd-Yusof, J., & Viswanathan, H. (2017). Predictions of first passage times in sparse discrete fracture networks using graph-based reductions. *Physical Review E*, *96*, 13304. https://doi.org/10.1103/PhysRevE.96.013304

Hyman, J. D., Jiménez-Martínez, J., Viswanathan, H. S., Carey, J. W., Porter, M. L., Rougier, E., et al. (2016). Understanding hydraulic fracturing: A multi-scale problem. *Philosophical Transactions of the Royal Society A*, *374*(2078), 20150426.

Hyman, J. D., & Jiménez-Marttínez, J. (2018). Dispersion and mixing in three-dimensional discrete fracture networks: Nonlinear interplay between structural and hydraulic heterogeneity. *Water Resources Research*, *54*, 3243–3258. https://doi.org/10.1029/2018WR022585

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 & Geosciences*, *84*, 10–19.

Hyman, J. D., Painter, S. L., Viswanathan, H., Makedonska, N., & Karra, S. (2015). Influence of injection mode on transport properties in kilometer-scale three-dimensional discrete fracture networks. *Water Resources Research*, *51*, 7289–7308. https://doi.org/10.1002/2015WR017151

Jordan, A. B., Stauffer, P. H., Zyvoloski, G. A., Person, M. A., MacCarthy, J. K., & Anderson, D. N. (2014). Uncertainty in prediction of radionuclide gas migration from underground nuclear explosions. *Vadose Zone Journal*, *13*(10).

Joyce, S., Hartley, L., Applegate, D., Hoek, J., & Jackson, P. (2014). Multi-scale groundwater flow modeling during temperate climate conditions for the safety assessment of the proposed high-level nuclear waste repository site at Forsmark, Sweden. *Hydrogeology Journal*, *22*(6), 1233–1249.

Kang, P. 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*, *92*(2), 22148.

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*, 8646–8657. https://doi.org/10.1002/2014WR016829

Karra, S., O'Malley, D., Hyman, J., Viswanathan, H., & Srinivasan, G. (2018). Modeling flow and transport in fracture networks using graphs. *Physical Review E*, *97*(3), 33304.

LaGriT (2013). Los Alamos grid toolbox, (LaGriT). http://lagrit.lanl.gov, last Checked : July 14, 2017.

Lichtner, P. C., Hammond, G., Lu, C., Karra, S., Bisht, G., Andre, B., et al. (2015). PFLOTRAN user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes (Technical Report, (Report No.: LA-UR-15-20403)). Los Alamos, NM: Los Alamos National Laboratory.

Lin, Y., Le, E. B., O'Malley, D., Vesselinov, V. V., & Bui-Thanh, T. (2017). Large-scale inverse model analyses employing fast randomized data reduction. *Water Resources Research*, *53*, 6784–6801. https://doi.org/10.1002/2016WR020299

Maillot, J., Davy, P., Le Goc, R., Darcel, C., & de Dreuzy, J.-R. (2016). Connectivity, permeability, and channeling in randomly distributed and kinematically defined discrete fracture network models. *Water Resources Research*, *52*, 8526–8545. https://doi.org/10.1002/2016WR018973

Makedonska, N., Hyman, J. D., Karra, S., Painter, S. L., Gable, C. W., & Viswanathan, H. S. (2016). Evaluating the effect of internal aperture variability on transport in kilometer scale discrete fracture networks. *Advances in Water Resources*, *94*, 486–497.

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. *Computational Geosciences*, *19*(5), 1123–1137.

Mustapha, H., & Mustapha, K. (2007). A new approach to simulating flow in discrete fracture networks with an optimized mesh. *SIAM Journal on Scientific Computing*, *29*, 1439.

National Research Council (1996). *Rock fractures and fluid flow: Contemporary understanding and applications*. Washington, DC: National Academies Press.

Neuman, S. (2005). Trends, prospects and challenges in quantifying flow and transport through fractured rocks. *Hydrogeology Journal*, *13*(1), 124–147.

Noetinger, B., & Jarrige, N. (2012). A quasi steady state method for solving transient dDarcy flow in complex 3D fractured networks. *Journal of Computational Physics*, *231*(1), 23–38.

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

O'Malley, D., Karra, S., Hyman, J. D., Viswanathan, H. S., & Srinivasan, G. (2018). Efficient Monte Carlo with graph-based subsurface flow and transport models. *Water Resources Research*, *54*, 3758–3766. https://doi.org/10.1029/2017WR022073

O'Malley, D., & Vesselinov, V. V. (2014a). Groundwater remediation using the information gap decision theory. *Water Resources Research*, *50*, 246–256. https://doi.org/10.1002/2013WR014718

O'Malley, D., & Vesselinov, V. (2014b). A combined probabilistic/nonprobabilistic decision analysis for contaminant remediation. *SIAM/ASA Journal on Uncertainty Quantification*, *2*(1), 607–621.

Painter, S., & Cvetkovic, V. (2005). Upscaling discrete fracture network simulations: An alternative to continuum transport models. *Water Resources Research*, *41*, W02002. https://doi.org/10.1029/2004WR003682

Painter, S., Cvetkovic, V., & Selroos, J.-O. (2002). Power-law velocity distributions in fracture networks: Numerical evidence and implications for tracer transport. *Geophysical Research Letters*, *29*(14), 1676.

Painter, S. L., Gable, C. W., & Kelkar, S. (2012). Pathline tracing on fully unstructured control-volume grids. *Computational Geosciences*, *16*(4), 1125–1134.

Park, Y.-J., Lee, K.-K., Kosakowski, G., & Berkowitz, B. (2003). Transport behavior in three-dimensional fracture intersections. *Water Resources Research*, *39*(8), 1215. https://doi.org/10.1029/2002WR001801

Peherstorfer, B., Willcox, K., & Gunzburger, M. (2016). Optimal model management for multifidelity Monte Carlo estimation. *SIAM Journal on Scientific Computing*, *38*(5), A3163–A3194. https://doi.org/10.1137/15M1046472

Pichot, G., Erhel, J., & de Dreuzy, J.-R. (2010). A mixed hybrid mortar method for solving flow in discrete fracture networks. *Applicable Analysis*, *89*(10), 1629–1643.

Pichot, G., Erhel, J., & de Dreuzy, J.-R. (2012). A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks. *SIAM Journal on Scientific Computing*, *34*(1), B86–B105.

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, *6*, 1939–1959.

Rizzo, C. B., & de Barros, F. P. (2017). Minimum hydraulic resistance and least resistance path in heterogeneous porous media. *Water Resources Research*, *53*, 8596–8613. https://doi.org/10.1002/2017WR020418

Sævik, P. N., & Nixon, C. W. (2017). Inclusion of topological measurements into analytic estimates of effective permeability in fractured media. *Water Resources Research*, *53*, 9424–9443. https://doi.org/10.1002/2017WR020943

Santiago, E., Romero-Salcedo, M., Velasco-Hernández, J. X., Velasquillo, L. G., & Hernández, J. A. (2013). An integrated strategy for analyzing flow conductivity of fractures in a naturally fractured reservoir using a complex network metric. In I. Batyrshin, & M. G. Mendoza (Eds.), *Advances in computational intelligence: 11th Mexican International Conference on Artificial Intelligence, MICAI 2012, San Luis Potosí, Mexico, October 27 – November 4, 2012. Revised Selected Papers, Part II*. Berlin Heidelberg: Springer, pp. 350–361.

Santiago, E., Velasco-Hernández, J. X., & Romero-Salcedo, M. (2014). A methodology for the characterization of flow conductivity through the identification of communities in samples of fractured rocks. *Expert Systems With Applications*, *41*(3), 811–820. https://doi.org/10.1016/j.eswa.2013.08.011

Santiago, E., Velasco-Hernández, J. X., & Romero-Salcedo, M. (2016). A descriptive study of fracture networks in rocks using complex network metrics. *Computational Geosciences*, *88*, 97–114. https://doi.org/10.1016/j.cageo.2015.12.021

Silverman, B. (1982). Algorithm as 176: Kernel density estimation using the fast Fourier transform. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *31*(1), 93–99.

Stockman, H. W., Li, C., & Wilson, J. L. (1997). A lattice-gas and lattice Boltzmann study of mixing at continuous fracture junctions: Importance of boundary conditions. *Geophysical Research Letters*, *24*(12), 1515–1518.

Tyukhova, A. R., & Willmann, M. (2016). Connectivity metrics based on the path of smallest resistance. *Advances in Water Resources*, *88*, 14–20.

Valera, M., Guo, Z., Kelly, P., Matz, S., Cantu, V. A., Percus, A. G., et al. (2018). Machine learning for graph-based representations of three-dimensional discrete fracture networks. *Computational Geosciences*, *22*, 695–710. https://doi.org/10.1007/s10596-018-9720-1

Wellman, T. P., Shapiro, A. M., & Hill, M. C. (2009). Effects of simplifying fracture network representation on inert chemical migration in fracture-controlled aquifers. *Water Resources Research*, *45*, W01416. https://doi.org/10.1029/2008WR007025