Analysis and Visualization of Discrete Fracture Networks Using a Flow Topology Graph

Analysis and Visualization of Discrete Fracture Networks Using a Flow Topology Graph

Garrett Aldrich[1,2], Jeffrey D. Hyman[3,4], Satish Karra[3], Carl W. Gable[3],
Nataliia Makedonska[3], Hari Viswanathan[3], Jonathan Woodring[2], and Bernd Hamann[1]

Abstract—We present an analysis and visualization prototype using the concept of a flow topology graph (FTG) for characterization of flow in constrained networks, with a focus on discrete fracture networks (DFN), developed collaboratively by geoscientists and visualization scientists. Our method allows users to understand and evaluate flow and transport in DFN simulations by computing statistical distributions, segment paths of interest, and cluster particles based on their paths. The new approach enables domain scientists to evaluate the accuracy of the simulations, visualize features of interest, and compare multiple realizations over a specific domain of interest. Geoscientists can simulate complex transport phenomena modeling large sites for networks consisting of several thousand fractures without compromising the geometry of the network. However, few tools exist for performing higher-level analysis and visualization of simulated DFN data. The prototype system we present addresses this need. We demonstrate its effectiveness for increasingly complex examples of DFNs, covering two distinct use cases – hydrocarbon extraction from unconventional resources and transport of dissolved contaminant from a spent nuclear fuel repository.

Index Terms—Network Flow Analysis and Visualization, Flow Topology Graph, Topological Path Analysis, Topological Trace Clustering, Flow in Fractured Rock, Discrete Fracture Network

✦

1 INTRODUCTION

WE present a method for the analysis and visualization of constrained flow networks driven by applications in computational simulation of fluid flow and transport in fractured rock. Determining how subsurface fractures control flow and transport has various applications in engineering and scientific endeavors including hydrocarbon extraction, aquifer storage and management, geothermal energy extraction, environmental restoration of fractured rock contaminated sites and the disposal of spent nuclear fuel [1], [2], [3]. Figure 2 shows a caricature of flow and transport in fractured porous media.

Discrete fracture networks (DFN) are one methodology that computational subsurface scientists use to simulate fluid transport within such fracture networks. Although DFN models were introduced over two decades ago, they are fairly novel to the visualization community. The need for this collaboration between the visualization researchers and geoscientists is the result of recent developments in the DFN community where three-dimensional fracture networks consisting of tens of thousands of fractures are now common. Accumulating local and global transport statistics such as the distribution of traversal times, velocities, and tortuosity of advected particles is not terribly difficult, but the analysis of these flow features and determining their relation to properties of the simulation domain is a demanding task.

The large amount of data resulting from these physics based simulations has created a need for advanced analysis and visualizations techniques to more efficiently process and interpret model outputs. Our team has addressed this need by developing, implementing, and testing a new visualization workflow.

We identified three areas of analysis research needed by geoscientists concerned with flow and transport in fractured media: statistical analysis, topological path analysis, and topological trace clustering. Figure 1 shows an outline of the methodology we developed, which is described in this paper, to address these needs. Using particle trajectories in the DFN (Left) we build a flow topology graph (FTG) (Center Top) that embeds information about transport through the fracture network into a graph. This representation enables us to develop and use graph analytics based algorithms, which combine feature and statistical analysis, to analyze the simulation output. The analysis results are stored in the FTG and are used to generate geometry files and statistical plots which can be explored by the user (Right). Using this information, we evaluate and refine the FTG analysis to investigate specific features of flow and transport through each DFN (Center Bottom). Embedding analysis from the FTG directly into geometry files for the DFN allows for integration of the FTG data with standard visualization tools. Furthermore, these tools allow for the identification of backbones in the DFN, which are connected subsets of fractures on which a majority of flow and transport occurs.

The key contributions of this methodology development are:

- Advanced visualization for the analysis of fluid transport in DFN simulations.
- The direct transformation of the simulation results

- [1] *Institute for Data Analysis and Visualization, Department of Computer Science, University of California Davis, Davis, CA 95616*
- [2] *Data Science at Scale Division, (CCS-7), Los Alamos National Laboratory, Los Alamos, NM, 87544*
- [3] *Earth and Environmental Sciences Division, (EES-16), Los Alamos National Laboratory, Los Alamos, NM, 87544*
- [4] *Center for Nonlinear Studies, Theoretical Division, Los Alamos National Laboratory, Los Alamos National Laboratory, Los Alamos, NM, 87544*
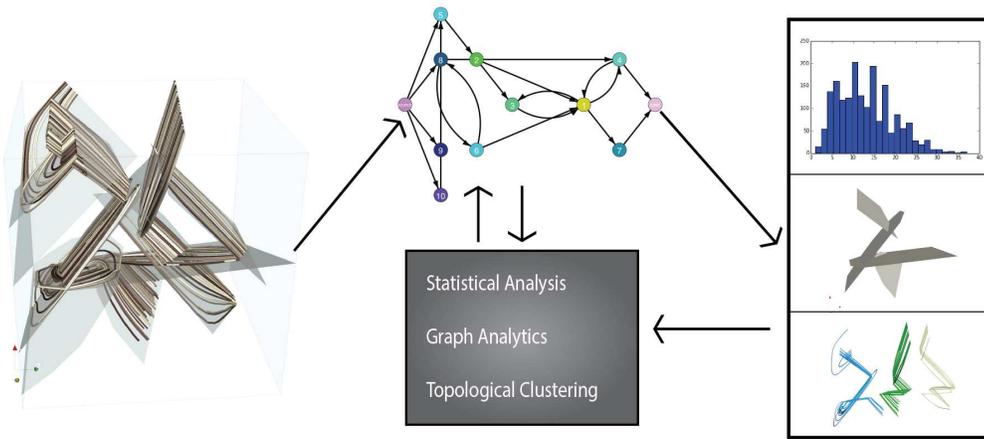
Fig. 1. Overview of our analysis and visualization methodology. (Left) Transport through a discrete fracture network (DFN) is simulated using particle advection through a steady-state flow field. (Center Top) Pathlines obtained in the transport simulation are used to construct a flow topology graph (FTG) that embeds flow and transport information into a graph. (Right) Analysis of the FTG using statistics, graph theory, and topological clustering provides detailed information about various features of interest. Using this information, we can evaluate and refine the FTG analysis to investigate specific features of flow and transport through each DFN (Center Bottom).

to a flow topology graph (FTG) with one-to-one correspondence with DFN geometry.

- A novel graph analysis algorithm for detecting backbone paths on FTG.
- A new algorithm for clustering particle trajectories based their path topology that can be used to identify and quantify flow channeling within the DFN.

In section 2, we motivate the need for these DFN analysis methods, provide a summary of relevant previous work, and briefly describe the DFN methodology. In section 3, we define the flow topology graph (FTG), the cornerstone of our analysis framework, and describe how it is constructed using transport simulations on a DFN. In section 4, we describe FTG based algorithms we developed for DFN analysis. In section 5, we demonstrate the effectiveness of these methods by applying them to several fracture networks that vary in terms of size and number of fractures. The least complex is a synthetic network consisting of two hundred fractures and the most complex is based on actual site data and contains about five thousand fractures. We provide conclusions in section 6, where limitations and future work are also discussed.

2 MOTIVATION AND BACKGROUND

Our primary goal is to create a robust methodology to analyze and visualize flow and transport in three-dimensional DFN models using advective Lagrangian particle-tracking. To accomplish this goal, it is necessary to have a flexible and scalable workflow because DFN vary greatly in domain size and fracture density. Statistical analysis is used to identify potential problems in the simulation and allows for better understanding of system-wide trends. Topological path analysis allows for the identification of important regions within the network. Specifically for DFN simulations, we want to identify backbones in the fracture network. We define a *backbone* as a connected subset of fractures in the network on which a substantial portion of flow and transport occurs; a DFN can have multiple backbones. Finally,
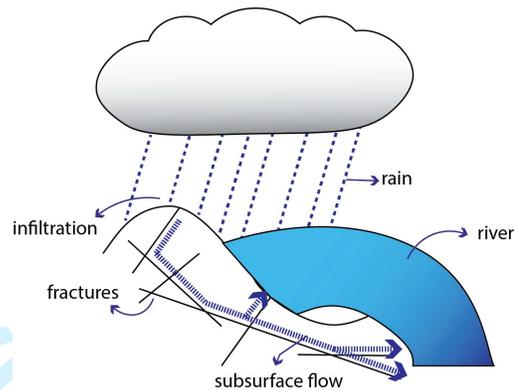
Fig. 2. A caricature of a two-dimensional fracture network embedded within impermeable rock. Fractures are the principal pathway for flow and transport through low-permeability rocks in the subsurface. Beyond the difficulties associated with determining flow and transport within such networks, efficient and effective ways for the analysis of the data sets produced via such simulations are still lacking. Existing general data analysis and visualization methods must be specialized for the needs of particle tracking through fracture network data.

topological trace clustering is used to identify groups of particles that travel along similar paths. These clusters provide insight into transport phenomenon, such as clustering and flow channeling, within the DFN and are used as a verification of backbones.

2.1 Related Work

Flow and transport in DFN is a new application in the visualization community – Laramee et al. [4], Salzbrunn et al. [5], and Pobitzer et al. [6] produced state-of-the-art reports that provide a good overall background. These references focus on the general analysis of flow fields and particle traces including topological analysis, feature selection, and streamline segmentation. In general, our problem can be posed as studying flow and transport in a constrained network. While the analysis goals were different, Jones and Ma [7] address a similar application by visualizing fluid particle

trajectories in porous media that are not constrained by fracture geometry. They used a warped curvature plot of the trajectories to allow the user to select particle traces of interest, and then visualize those trajectories while emphasizing local geometry. Another similar application is using fiber tracking to define topologically salient regions in diffusion tensor MRI data of the brain, which can be thought of as a constrained network flow problem [8], [9].

An increasingly common technique for analyzing flow, one that we use in this work, is to generate a graph structure derived from advected particles in a flow field. Xu and Shen. [10] built a graph structure, the flow web, by segmenting a flow field with a regular grid and building a graph from the connections made by advected particles in the field. They allow users to query the graph directly through a visual representation. This allows selected paths in the graph to be visualized as segmented streamlines in the flow field. Nouanesengsy [11] and C. M. Chen [12] used this graph structure to facilitate load balancing during parallel processing of very large flow fields. In contrast to these works we use the explicit network topology of a DFN to define our graph, which is then used for automated feature analysis and clustering. Flow graphs have also been defined from the topological properties of a vector field itself. For example, G. Chen et al. analyze flow by generating the Morse connection graph from the Morse decomposition of flow fields [13], [14], [15].

Clustering particle traces or other integral curves is a common method for dealing with occlusion and emphasizing flow behavior for a system. Similarities in streamline shape or geometric features are commonly used for clustering. McLoughlin et al. [16] introduced an efficient and interactive method for identifying similar streamlines by their curvature, torsion, and tortuosity. Yu et al. [17] used spatial and geometric data to cluster streamlines. Lu et al. [18] used these properties to define statistical distributions for streamline and combine querying with clustering for analysis. Wei et al [19] used a model-based clustering method for finding similarity in particle trajectories from turbulent combustion data to demonstrate the relationship between phase and physical space. They used a two-stage clustering algorithm. In the first stage, time series curves are clustered automatically, and then a user is able to update the clustering in a second pass after exploring the automated results.

2.2 Discrete Fracture Networks

DFN modeling is one approach to simulate flow and transport through low-permeability fractured rocks, such as shale or granite, in the subsurface. In this approach, geologic field investigations are used to create a network of fractures where the geometry and properties of individual fractures are explicitly represented as lines in two dimensions and planar polygons in three dimensions. Fractures in the network are assigned a shape, location, aperture, and orientation based on distributions determined by a geological survey. Once a network is constructed, the individual fractures are meshed for computation and the flow equations are numerically integrated on the resulting computational mesh. For a more detailed explanation of various DFN
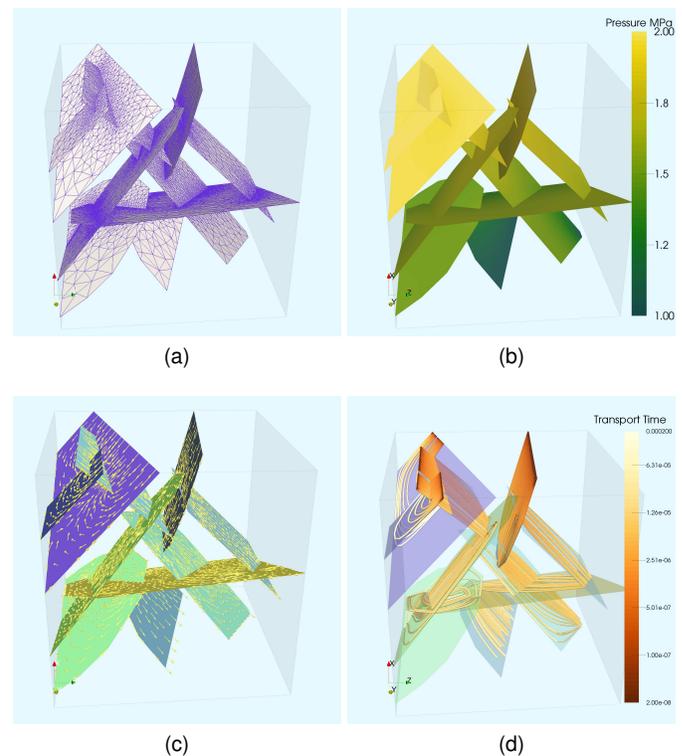
Fig. 3. DFN flow and transport example. This DFN is made up of ten interconnected fractures within a 10^3 meter domain. (a) The conforming Delaunay mesh overlaid on the fractures. (b) A steady-state pressure field in the network. Flow is primarily in the vertical direction. (c) The pressure solution is used to compute the fluid velocity field on each fracture. (d) Particles are inserted along fracture edges at the top of the domain. Particle trajectories define path-lines in the flow solution and identify where transport occurs in the system. Path-lines here are colored by transit time.

methodologies and their applications see [20], [21], [22], [23], [24], [25].

Figure 3 shows our workflow for simulating transport in a DFN. The DFNWORKS [26] computational suite is used to generate each DFN and resolve flow and transport therein. Each DFN is constructed using the feature rejection algorithm for meshing (FRAM) [27]. Then a conforming Delaunay triangulation, whose dual Voronoi mesh can then be used as the control volume for finite volume solvers, is created using the LAGRIT meshing toolkit [28], Figure 3 (a). The massively parallel flow solver PFLOTRAN [29] is then used to determine the steady-state pressure field for the DFN, Figure 3 (b). Darcy's law and the local pressure field are then used to reconstruct the local velocity field, Figure 3 (c), which is used to track particles through the domain [30], [31], Figure 3 (d). In the next section we describe our method for constructing a FTG using these particle trajectories, referred to as traces.

3 THE FLOW TOPOLOGY GRAPH

The flow topology graph (FTG) is a direct abstraction of flow and transport constrained to a discrete fracture network. We define the FTG for simulated particle transport on DFN using path traces in conjunction with the fracture network topology. Specifically, each fracture is represented by an individual vertex in the graph and edges represent the discrete
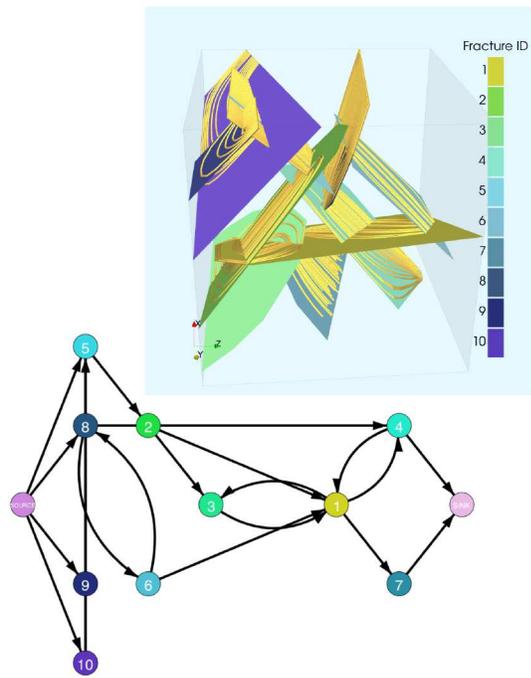
Fig. 4. The flow topology graph (FTG) derived from the transport simulation shown in Figure 3 (d). In the FTG each fracture polygon is represented by a vertex in the graph. Each edge represents particles that travel between two fractures over an intersection. We also add a "SOURCE" and "SINK" node (at the far left and right respectively) to the FTG from which all particles enter and exit.

Symbol	Description
F_i	Fracture number i in a DFN
V_i	Vertex in the FTG, V_i represents fracture F_i
$\mathcal{P}_j$	Particle j transported on a DFN
$|\mathcal{P}^i|$	Total number of particles which travel on F_i
$\mathcal{R}_j$	Ordered set of fractures particle $\mathcal{P}_i$ traverses
E_i	The minimum number of fractures a particle would have to traverse to exit the DFN from F_i
S_i	The minimum number of fractures a particle would have to traverse to enter F_i
L_j	The path length of the trace for $\mathcal{P}_j$
C_j	The net displacement of $\mathcal{P}_j$
T_j	The tortuosity of the trace for $\mathcal{P}_j$
$\bar{T}^i$	The mean tortuosity of all particles starting at F_i until they exit the DFN

TABLE 1
Reference table of symbols used to define the flow topology graph for discrete fracture networks.

intersections where particles can transport from one fracture to another. The FTG encapsulates information of particles traversing the network, which have local driving forces incorporated. Therefore, these driving forces, e.g., gravity and local pressure gradient determine the topological order and are implicitly included in the graph.

Formally, for a DFN with N fractures ($F_i : i = 1, \ldots, N$) and K particles ($\mathcal{P}_j : j = 1, \ldots, K$) we build a FTG using the following algorithm. First, we create N vertices $V_i : i = 1, \ldots, N$, each corresponding to an F_i. We include *source* (V_{source}) and *sink* (V_{sink}) vertices that represent the global entry and exit into the system for particles. Each trace path ($\mathcal{R}_j$) contains an ordered set of fracture polygons that are visited by $\mathcal{P}_j$. For every entry fracture, the first fracture F_u in every trace path $\mathcal{R}_j$, we create an edge from V_{source} to V_u. Similarly for every exit fracture, the final fracture F_w in every $\mathcal{R}_j$, we create an edge from V_w to V_{sink}. Each time a particle transitions from one fracture, F_u, to another, F_w i.e., there are ordered pairs (F_u, F_w) in $\mathcal{R}_j$, we create an edge from V_u to V_w. Multiple edges directed from the same vertices are merged, so at most a single directed edge exists from any vertex V_u to any V_w. For each vertex V_i we append a pointer to every particle, $\mathcal{P}_j$, that traverses the fracture polygon, F_i. A pointer to every particle that traverses an intersection between two fractures is also added to the associated edge. Table 1 summarizes our notation for reference.

Figure 4 shows the FTG for transport in the ten fracture networks shown in Figure 3 (d) where the fracture IDs and colormap show the correspondence to graph vertices. For the purpose of this example, we lay out the vertices by

their topological distance from the source (left most vertex) and sink (right most vertex), such that columns of vertices have the same topological distance. It is important to note that this layout is used for the purposes of this paper, and does not scale to large DFN. We do not currently directly visualize the graph as part of our analysis framework. Directed edges between vertices indicate that at least one particle passed between the corresponding fractures in the direction indicated by the edge.

One benefit of the FTG is that adding information about the specific flow network, model geometry, and transport statistics to vertices in the network is straightforward. The first particle based attribute we append to the vertices is the number of particles that pass through each fracture. Another is the topological distance to the exit, E_i, which is the minimum number of fractures a particle must traverse before exiting the network from F_i. We also calculate the topological distance from the source, S_i, which is the minimum number of fractures a particle must traverse before entering F_i. We derive per-vertex and per-edge statistics for the particles that traverse the associated fracture and fracture intersections respectively. For example, we compute of the tortuosity of each trajectory (the ratio between the path length of a trace and the net displacement of the particle) which measures the deviation of a trajectory from the straight line between its edge points. Formally, we define the fracture tortuosity, T_j^i, of a particle $\mathcal{P}_j$ for fracture F_i as:

$$T_j^i = L_j^i / C_j^i,$$

where L_j^i is the length of the trace of $\mathcal{P}_j$ starting from the point it enters fracture F_i until it exits the DFN. Similarly C_j^i is the Euclidean distance from the two end points taken over the same interval. Tortuosity can be defined multiple ways, but this definition is adequate for our purposes.

4 ANALYSIS AND VISUALIZATION OF DFN

Using the FTG representation, we provide a workflow that produces three types of analysis products for DFN, namely statistical analysis, path analysis, and topological clustering. Statistical analysis can be used for debugging and to

compare multiple fracture networks based on the same geological distributions to address questions of ergodicity and resolve global and local trends in the flow field. Path analysis allows us to use a Lagrangian viewpoint to find features in the flow field and link them to the fracture geometry. One key feature of interest within a DFN are backbones, which are connected subsets of fractures on which a majority of flow and transport occur. Backbones are believed to be responsible for flow channelization in fractured media, where flow is concentrated in certain regions of fractured rock, and have been qualitatively identified [23], [32]. In this section, we provided a systematic methodology to identify them using the FTG. We describe a method of topological clustering to group particles which take similar paths through the network. This clustering allows us to better visualize and segment particle traces as well as verify backbones defined using path analysis.

4.1 Statistical Analysis

The construction of each DFN is stochastic, relying on randomly sampling known distributions of fracture size, orientation, aperture and shape; multiple realizations of a given site must therefore be created. Statistical analysis can be used to verify that an ensemble of fracture networks with different topologies, but modeling the same formation of rock, produce similar results. Sampling constraints in both time and space limit what experiment data can be obtained in the field; local measurements of key phenomena are not possible throughout a site. Therefore, upscaled quantities, are used for verification of flow and transport simulations at site specific locations. Statistical analysis can also be used to compare transport behavior on selected sets of fractures or paths (such as backbones) to global transport for the system.

By appending fracture and intersection attributes as well as statistics to the FTG for a transport simulation, we can readily accumulate global statistics for a single DFN, a local subset of the DFN, or multiple realizations. While global statistics are important for comparison between transport simulations, localized statistics taken from a subset of the network are useful for characterizing specific flow attributes. For example, the user can segment the parts of the network that are never reached by particles, find the set of fractures responsible for the fastest or slowest transport times, or segment the DFN into topological layers by finding all fractures where particles must travel through at least N fractures before entering.

Several attributes are stored in the FTG that are of interest to the domain experts analyzing the transport simulations of a DFN. Per-fracture attributes include the size; topological distances (E^i, S^i); the number of transported particles ($|\mathcal{P}^i|$); and mean fracture tortuosity ($\bar{T}^i$). For each particle, we store both per-fracture and total transport time, path length, velocity, and particle tortuosity (T_j). For these integral values, we are also interested in how they change as particles traverse the network. To accomplish this investigation, we parameterize the derived values for each particle over time, trace-path length, and topological distance. For example, in Figure 5 the tortuosity is shown for all particles in a simulation that uses the DFN shown in Figure 3. In the top plot of Figure 5, the maximum path length for all particle
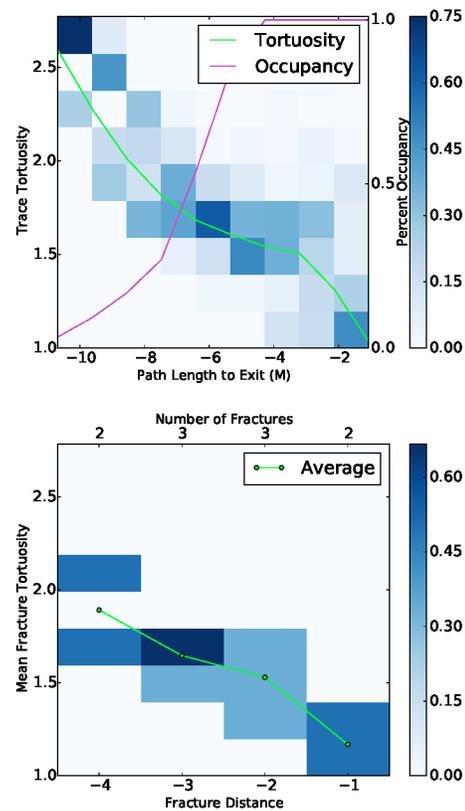
Fig. 5. Plots of statistical qualities from a transport simulation in a DFN made up of ten fractures. Around 150 particles are advected through the DFN to produce these statistics. (Top) A blue heat map represents the two-dimensional histogram of tortuosity values sampled at discrete path lengths along particle traces (blue gradient), as well as the mean curve (green). The particle occupancy, or number of particles in the system at each sampled point is also shown in magenta. The coarseness of the plot is due to the simplistic nature of our example. (Bottom) Particle tortuosity distribution are sampled over topological distance indicating that as particles reach fractures closer to the exit, they take more direct, less tortuous, paths.

trajectories is calculated L_{max}, and then divided evenly by the number of sample points. We parameterize each trajectory and sample the tortuosity starting at these points along the curve, discarding trajectories that have a shorter overall path length than the sample point. The result is a two-dimensional histogram, which we display as a heat map, in shades of blue. In addition, the mean tortuosity curve is plotted in green, and the number of particles sampled at that point (the occupancy) is plotted in magenta. When plotting the results we add a negative value to the distances that indicates the path length to the exit, as apposed to from the source. We can interpret from these plots that the farther a particle travels through the DFN, the more direct its path towards the exit becomes. In the bottom plot, the change in fracture tortuosity over topological distance to the exit, E^i, is shown. This parameterization is necessarily much coarser, and the number of fractures at each distance is given for reference. From this graph we show that fractures closer to the DFN exit, provide a more direct path.

Most often the statistical data is displayed by producing plots and graphs, however we also allow the user to directly
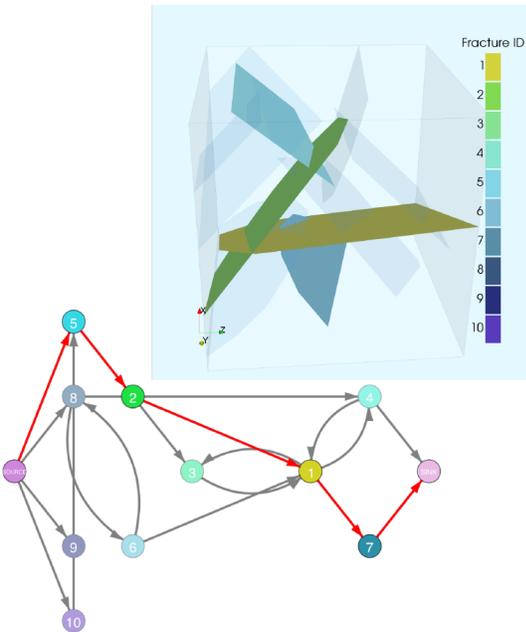
Fig. 6. Path analysis in the flow topology graph enables the extraction of features within the flow network such backbones. These backbones form where a significant portion of flow becomes channelized. The path is highlighted on the original geometry of the discrete fracture network as well as being represented by the FTG.

visualize these distributions on the original geometry. For large DFN, rendering the entire network using a colormap to identify attributes is less useful, due to occlusion. However, this method of visualization is still relevant when applied to meaningful subsets of the DFN.

4.2 Path Analysis

Flow channeling and fast paths through fractured media lead to non-Fickian (anomalous transport) [3], [33]. Geoscientists and hydrologists are interested in characterizing the features of a fracture network that result in these flow charachteristics. Within the context of DFN simulations, we are specifically interested in identifying paths in the FTG, connected sets of fractures, where the local flow magnitude is significantly above the network average. We refer to these connected sets of fractures as *backbones* because they act as the primary transport path through the network. For a candidate path to be a backbone it must meet the following criteria: (i) Backbones carry a large percentage of the localized flow, (ii) Particles near or on a backbone should remain on that path rather than dispersing throughout the network, and (iii) Backbones are be composed of a relatively small number of fractures that provide a direct path towards an outlet. If candidate paths meet these requirements, then they will be independently verified using the particle clustering algorithm discussed in section 4.3. Clustering of particles is indicative of flow channeling and allow one to determine whether or not a path is a backbone. A DFN may have multiple backbones and backbones in a network are not unique, e.g., they may depend on the imposed principal direction of flow [33]. Backbones have been qualitatively identified in DFN simulations as discussed in the hydrology

literature [23], [32]. In this subsection, we provide a data-driven algorithm based on trace-path analysis to identify and characterize backbones.

To extract candidate paths we consider these three criteria on the FTG representation of the DFN. Then the problem becomes equivalent to extracting shortest paths from the FTG. Figure 6 shows an example of a channelized path extracted from the FTG and the resulting set of fractures which represent that path in the DFN. This reduction requires us to define edge weights such that a path corresponds to a candidate backbone when it has been extracted as a shortest path. We have tested several edge weights and obtained the best results with the following:

$$W_1(e_{i,j}) = \bar{T}^j, \tag{1}$$

$$W_2(e_{i,j}) = \bar{T}^j * \sqrt{E_j}, \tag{2}$$

$$W_3(e_{i,j}) = -|\mathcal{P}^j|, \tag{3}$$

In (1) each edge, $e_{i,j}$, is weighted by the mean fracture tortuosity, $\bar{T}^j$, for the fracture represented by vertex j in the FTG. Lower fracture tortuosity implies that particles traveling along that fracture exit the system using a relatively straight path. We derive this metric from the fact that particles that are not on a backbone tend to disperse throughout the network, while particles on a backbone remain on it and exhibit a straighter trajectory. The second function, (2), combines (1) with the topological distance term, E_j, so that fractures which are closer to the exit have a lower weight. This metric produces paths similar to those obtained via (1), but reduces the influence of fractures that are near the entrance to the system. This reduction accounts for potential bias caused by initial seeding of particles that can influence backbone detection. Often particles traverse several fractures before traveling along a backbone path. By reducing the influence of entry fractures, we produce more meaningful backbones at the interior of the system. In practice paths are extracted either using (1) or (2) but not both, depending on a particular application and whether the influence of initial particle location is desired or not.

In the third function, (3), weights are the negative values of the total numbers of particles traversing a fracture pointed to by each edge. The weight is negative so that the optimal shortest path maximizes the number of particles. While (3) defines the most obvious candidate for a weight function, it is also the most difficult to implement. Specially, finding the optimal path for a general FTG with negative weights is intractable for large networks (assuming $P \neq NP$). Another consideration is ensuring that candidate backbone paths extracted from this method are not arbitrarily long, as each added fracture in the path increases the total number of particles. We resolve both problems by reducing the FTG into a directed acyclical graph FTG_{DAG}, which removes all cycles from the graph. The FTG is reduced to FTG_{DAG} by removing all edges where the topological distance to the exit increases or remains the same, i.e., the edges of FTG_{DAG} are the subset of edges such that $e_{j,k} : E_j > E_k$. This removes cycles (all of which would be negative) from the graph and allows for negative edge weights to be used. It also removes all edges that allow particles to traverse farther from the exit in topological

(a)   (b)
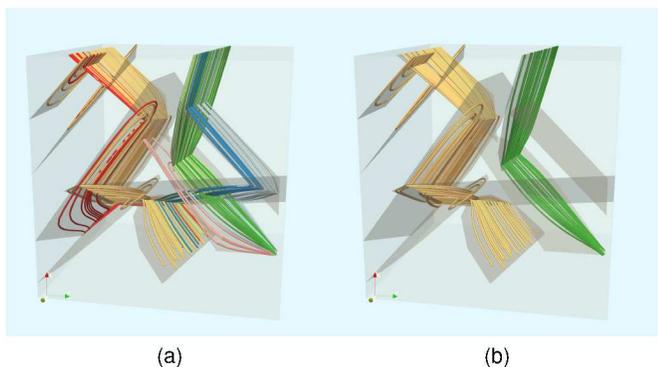
Fig. 7. We use agglomerative hierarchal clustering to segment particles which take similar topological paths through the network. For this DFN the clustering is readily apparent, and each cluster is rendered with a unique color. In (b) we show the largest two clusters emphasizing that clustered particles take similar paths while traversing the DFN.

distance, which eliminates arbitrarily long paths. Extracting paths using edge weight (3), produces candidate backbones that have a high number of particles, on direct paths that contain a limited number of fractures.

To account for multiple backbones in a DFN, multiple locally optimal paths are extracted by combining two methods. First, we extract paths between the source vertex of the FTG and every vertex representing an exit fracture where a significant percentage of particles leave the system. This approach is apposed to only extracting paths between the source and sink vertex in the FTG, which would produce a globally optimal solution. The minimum percentage of exiting fractures is a user-defined parameter; a value of three percent is used by default. However, the type of simulation, the number of exit fractures, and the initial seeding of particles all influence how this parameter is set. Second, we use a standard implementation of Yen's K shortest paths algorithm [34] which produces up to K optimal paths between two vertices in a graph. The choice for K is application-dependent; users must take into account the type of simulation, domain size, number of fractures, and the sizes of those fractures. In practice a value of $K = 3$ is used for most networks while $K = 8$ shortest paths have been needed for very large complex DFNs. The first method is necessary to extract multiple disjoint backbone paths, especially when they are of varying strength. The second is necessary for finding multiple intersecting backbones which exit through the same fracture.

To aid users in further visual analysis of the backbone candidates, we calculate the number of particles which travel along each path. This indicates the relative strength of each candidate. However, further validation is usually necessary to ensure channelization is occurring, which can be better shown through clustering particle traces by their path topology.

4.3 Topological Trace Clustering

Visually differentiating large numbers of integrated paths leads to an occlusion problem. A common approach to resolve this issue is to compare traces using a similarity metric and then apply a clustering algorithm to associate them with groups in accordance with their similarity. Once split into groups, particle traces can be visualized either through rendering each group in a different color, by rendering a smaller but representative subset of the particles, or a combination of both techniques. This allows users to better understand the coherency between particle traces and observe trends in the flow field or network. Previous research has focused on defining similarity metrics by properties of the curves themselves, such as curvature [16], shape [17], or statistical distributions [18]. However, similarity in shape or structure of individual trajectories is less important for our application.

We define a similarity function using network topology in terms of the ordered set of fractures each trace traverses. By representing the trace path of a particle using the ordered set of fractures that it visits while traversing a DFN, and comparing the paths each trace takes through the FTG the similarity of traces can be readily computed. Explicitly, any two traces that travel on the same ordered set of fractures are considered topologically equivalent when clustering. Similarly, if two particles have nearly the same trace path, deviating only slightly in the fractures they traverse, then they will be considered to have small distance value between them. Finally, if two particles have completely different trace paths then they will be assigned a very large or infinite distance.

This trace distance function is inspired by the Levenshtein distance function for string-based comparisons [35]. The Levenshtein distance function finds the shortest edit distance between two strings by recursively comparing the ordered set of characters in the string and produces the minimum number of changes needed to convert one string into the other. The possible changes include insertion, removal or replacement of single characters. For example, when comparing 'skip' and 'sip' or 'show' and 'slow', the Levenshtein distance is one in both cases (a removal and a replacement, respectively). Wilson et al. [36] used an adaptation of this metric to compare spectral representations of graphs. We have adapted this algorithm by considering strings of fracture IDs, F_i, representing the trace paths of particles, rather than strings of characters. For example, a particle could have the trace path, $\{F_1, F_3, F_{10}\}$, which would indicate that it entered the DFN on fracture F_1, was transported to fracture F_3 and exited the DFN through fracture F_{10}. A pseudocode implementation of the distance function is given in Algorithm 1.

The major difference between our algorithm and the original is the use of a topological cost function for making edits; $\phi(i, j)$ in Algorithm 1. In the Levenshtein distance, all edits are given a cost of 1, while we calculate the cost of replacing fractures in a trace path using the topological distance information encapsulated by the FTG. We define the cost function for replacing a fracture, F_i with another fracture F_j, $\phi(i, j)$, to be the number of edges in the shortest path from vertex v_i to v_j in the associated FTG. In other words, the replacement cost is represented by the minimum number of fractures that would be traversed for a particle on F_i to reach F_j. If no path exists in the FTG, then $\phi(i, j) = $ infinity. This definition also implies that the cost of insertion or deletion of a fracture is 1. This can be explained as follows: if there existed two trace paths, $\mathcal{R}_a = \{F_a, F_b, F_c\}$ and $\mathcal{R}_b = \{F_a, F_c\}$, then the FTG

would have to contain edges $e_{a,b}$, $e_{a,c}$. Therefore the cost to either remove F_b from $\mathcal{R}_a$ or add it to $\mathcal{R}_b$ is 1, as that is the minimum distance between the associated vertices. The resulting algorithm uses a recursive function that returns the minimum cost.

Algorithm 1 Trace Distance Function

Let $\mathcal{R}_a$ be the ordered set of fracture ids, F_i, for trace a
Let $|\mathcal{R}_a|$ be the number of fracture ids in $\mathcal{R}_a$
Let $\mathcal{R}_a|_k$ be the kth fracture id
Let $\phi(F_i,F_j)$ be the cost function for replacing IDs s.t.
$\quad \phi(F_i,F_j) \equiv$ the shortest path from v_i to v_j in the FTG

Initially $La \leftarrow |\mathcal{R}_a|$
Initially $Lb \leftarrow |\mathcal{R}_b|$
procedure TDF($\mathcal{R}_a$,La,$\mathcal{R}_b$,Lb)
$\quad$ **if** $La = 0$ **then**
$\quad\quad$ **return** Lb
$\quad$ **end if**
$\quad$ **if** $Lb = 0$ **then**
$\quad\quad$ **return** La
$\quad$ **end if**
$\quad$ **if** $\mathcal{R}_a|_{La} = \mathcal{R}_b|_{Lb}$ **then**
$\quad\quad$ $RC \leftarrow 0$
$\quad$ **else**
$\quad\quad$ $RC \leftarrow$ MIN($\phi(\mathcal{R}_a|_{La},\mathcal{R}_b|_{Lb})$,$\phi(\mathcal{R}_b|_{Lb},\mathcal{R}_a|_{La})$)
$\quad$ **end if**
$\quad$ **return** MIN(
$\quad\quad$ TDF($\mathcal{R}_a$,$La - 1$,$\mathcal{R}_b$,Lb) $+1$,
$\quad\quad$ TDF($\mathcal{R}_a$,La,$\mathcal{R}_b$,$Lb - 1$) $+1$,
$\quad\quad$ TDF($\mathcal{R}_a$,$La - 1$,$\mathcal{R}_b$,$Lb - 1$) $+RC$
$\quad\quad$)
end procedure

To identify trace clusters given their mutual distances, we use the agglomerative hierarchical clustering (AHC) algorithm [37]. This is one of the most commonly used method for clustering path traces and other integral curves, [5], [6], [16], [17], [18]. AHC builds a hierarchy by recursively merging pairs of clusters (initially each trace being its own cluster), until all clusters are merged. The resulting hierarchy can then be 'cut' by setting a maximum distance value for pairs of particle traces included in the same cluster. This gives the user control to define how similar the paths of particles must be. In practice we choose several distances and allow the user to select the most appropriate one during exploratory visualization.

In Figure 7(a) we show the results of our clustering algorithm for the ten-fracture system from section 2. Here, each cluster is rendered in a unique color. In Figure 7(b) we show only the two largest clusters to emphasize the paths taken by traces in each group. In particular the largest (yellow) cluster consists of particles that take similar, but not the same, path through the network. We note that the largest clusters lie primarily along the *backbone* path shown in Figure 7. By grouping and then visualizing large clusters of particles that take similar paths through the network, users are better able to observe and differentiate areas where channelization occurs. If a larger number of clusters is seen, then a single representative trace of each cluster will be rendered to further reduce occlusion retaining key information.
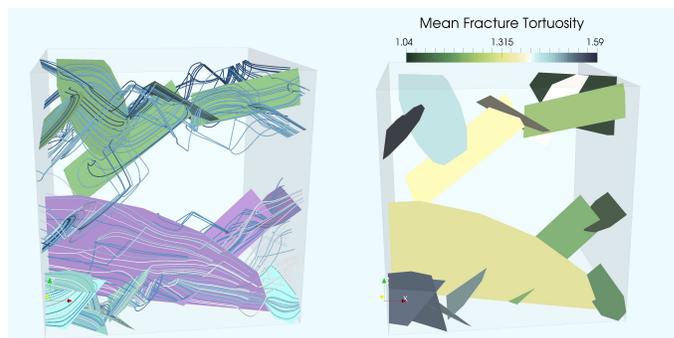
Fig. 8. Three backbones in a 200 fracture network. On the left, each path is represented by a different color and the clustered particles traveling along each path are represented by tubes whose colors indicate their cluster ID. On the right, backbone fractures are colored by the mean tortuosity of the particles traversing these fractures (fracture tortuosity). The tortuosity values are close to one indicating that particles on the backbones take direct paths towards the exit, rather than dispersing throughout the network.

The representative trace is chosen at random from particles which take the most common path in the cluster.

While our algorithm on its own performs well and produces clusters as expected, we have included two optional modifications to accommodate specific needs of DFN simulations. First, we allow users to disregard cycles in the fracture path by collapsing them in the trace path i.e., $\{\ldots,F_a,F_b,F_c,F_d,F_b,F_e,\ldots\} \rightarrow \{\ldots,F_a,F_b,F_e,\ldots\}$. In the context of a DFN, this corresponds to a particle that leaves a fracture, but later then returns to it and continues to travel therein.

The second modification accounts for potential bias due to initial conditions. To do so, we disregard the first several (typically 1 to 3) fractures in the trace path when computing clusters. Disregarding the first fractures in a trace path limits the influence that the initial seeding has on clusters by allowing particles to initially disperse/coalesce in the system prior to being subject to analysis. Hyman et al. [23] observed that is took particles uniformly distributed across an inlet plane 250 meters before they exhibited strong flow channeling characteristics. Trimming initial fractures is an optional step in the clustering algorithm and only useful for certain use cases and under certain initial conditions. However, it provides flexibility for domain experts. The motivation behind this option is similar to the reason used for our addition of topological distance when extracting candidate backbone paths, cf. equation (2).

By visualizing the large clusters of particles, channelization becomes more apparent. This is an important tool for validating the candidate backbones and observing the global behavior of the flow network.

4.4 Analysis Driven Visualization

Exploratory visualization of the analysis results produced by our framework plays an integral role in evaluating the results of a DFN simulation. We have chosen to decouple visualization from the analysis process to maintain both flexibility and interactivity. Statistical plots are produced directly from the FTG using python scripts that generate
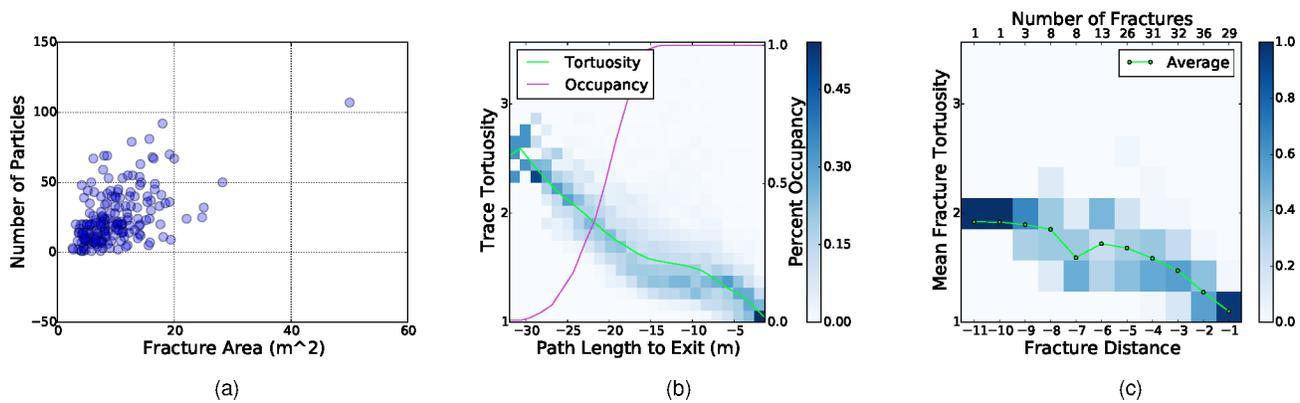
Fig. 9. We produce statistical results based on the particle behavior in a 200 fracture DFN with around 500 advected particles. The scatter plot in (a) shows the number of particles that traverse through fractures of different sizes. This image indicate that particles are well distributed over fractures of different sizes, except for a single outlying fracture that is significantly larger than the rest on which a large portion of the transport occurs. We calculate a series histogram for the tortuosity by sampling particles at regular intervals of time, path length (b), or mean distance and topological distance (c).

plots either for the entire system, a subset of fractures representing candidate backbones, or sets of clustered particles. These plots can be combined and overlaid for comparison purposes. This is especially useful for comparing multiple simulated data sets. While we provide predefined python-based scripts to produce these plots, custom plots can also be created. Direct geometric visualization is handled through embedding analysis results directly into geometry files for both the DFN and particle traces. The one-to-one mapping between the FTG and DFN allows us to add per-fracture attributes to the DFN geometry files. Similarly, the statistical and clustering information for each particle is added as an attribute to the trace geometry files. To allow users to explore the results of parameter value changes, we sample a selected parameter space and embed all of the results into the output files. For example, when generating clusters we select multiple minimum distances to cut the agglomerative clustering hierarchy and allow users to select from clusters generated at each level. The number of cuts and minimum distance are user-defined options. By default the mean distance between particles is used as a base. Five subsequent cuts are also made by linearly sampling between the mean distance and one-tenth that distance. The candidate backbones are embedded in two different ways, to aid users in identifying the most meaningful ones. Paths can be selected by the method and order that they are extracted or by the amount of flow occurring on each. The former allows users to better understand why the algorithm produced each candidate path and the latter gives a more natural ordering of the candidates. Individual traces and each fracture on a selected path retain their statistical properties along with cluster information. This allows each feature to be compared visually and the selections can be used to generate plots for the subset.

The decision to decouple visualization from analysis, rather than integrating both steps into a custom tool, makes possible the use of many standard visualization tools, such as the Paraview data analysis and visualization platform [38]. Our target user group, computational geoscientists studying flow in DFN simulations, preferred

to leverage preexisting tools that they are already familiar with. By using well-maintained visualization tools, we ensure easy use and that the analysis tools can remain usable without support for a new software interface. Furthermore, by encapsulating the analysis methods in an offline process, large amounts of data can be processed. As the field continues to develop, DFNs are expected to become larger, more complex and are likely to require and increasing number of realizations to capture properties of the stochastic system.

We have a minimal set of requirements for visualization tools to effectively visualize the analysis files produced by our system. The first, and most important, is the ability to select subsets of data by setting thresholds for embedded attributes. This enables users to select particle traces by the cluster they belong to or the size of the cluster, and the particular cut in the hierarchy that the clusters are produced from. Thresholding also allows users to select candidate backbones by the method used and order in which they are extracted, or by the amount of flow occurring on each path. The second requirement is to overlay multiple geometry files using the absolute positions of vertices. Finally, the tool must allow users to apply colormaps to the geometry which corresponds to embedded per-cell attributes. Other features that we use to generate the examples provided in this paper, though not necessarily required for analysis, include: rendering lines as tubes of varying thickness and assigning glyphs to represent points and vectors.

5 EXAMPLES

We demonstrate our methodology by analyzing the flow and transport in DFN simulations at various scales. We begin with a medium sized DFN made up of two hundred fractures. Then we demonstrate our approach in two sub-surface applications with networks made up of thousands of fractures. The first of these is used to study unconventional hydrocarbon extraction based on a shale formation in the Tuscaloosa, Alabama. The second DFN model is loosely based on a subset of fractures in Forsmark, Sweden (a potential host location for spent civilian nuclear fuel). We

(a)   (b)   (c)
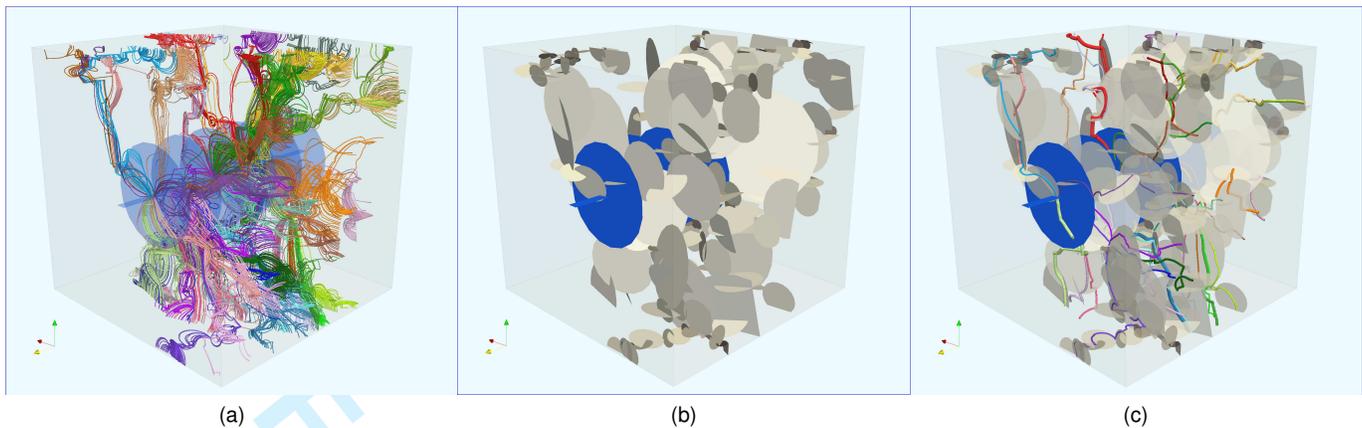
Fig. 10. A model of hydrocarbon extraction via hydraulic fracturing is represented by a DFN of a cubic kilometer of shale with a horizontal well (blue rectangular fracture in the center of the domain) and six larger fractures that represent the effects of hydraulic fracturing (blue circular fractures). In this example, flow is driven by drawdown pressure at the well. Figure (a) shows the large clustered particles for the system. Particles are not seeded on the back face. The large number of clusters indicate that channelization is occurring throughout the network. Backbones in the network are shown in (b), with representative particle traces from each cluster overlaid in (c). The thickness of each tube reveals the relative number of particles in each cluster, and the color remains consistent with the representation of these same clusters in (a).

selected these two site characterizations because they highlight different transport scenarios. The primary direction of flow in the hydrocarbon extraction model is radial, towards a horizontal well at the center of the domain, while in the DFN based on the Forsmark site, the imposed pressure gradient drives flow in one primary direction aligned with the Z-axis. We also use the method to compare multiple DFN realizations based on the same statical distributions.

5.1 Two hundred fracture network

We created a medium sized DFN of two hundred circular and rectangular fractures constrained to a 12 meter cubed domain. This network is used to demonstrate how the combination of statistical and visualization analysis allows us to characterize transport behavior for DFN. A pressure gradient is applied along the X-direction to create flow through the network and about 500 particles are used to simulate fluid transport.

Figure 8 shows three backbones identified in the network along with particle trajectories. Backbones are large fractures aligned with principal direction of flow. The backbone along the bottom of the domain, colored purple, is primarily a single large fracture while the other two are composed of several fractures. On the left, each path is represented by a different color and the clustered particles traveling along each path are represented by tubes whose colors indicate their cluster ID. On the right, fractures are colored by the mean tortuosity of particles on those fractures. The tortuosity values are close to one, indicating that particles traveling on these paths take a direct path towards the exit rather than dispersing into the rest of the network.

Figure 9 shows various particle based observables. Figure 9(a) shows a scatter plot of particle density as a function of relation to the fracture area and reveals that a large percentage of transport in the network occurs on a single large fracture. Fracture radii are determined by sampling a power law distribution so there are a lot of small fractures and few large ones. There are a disproportionate number

of particles on the largest fracture(s) when compared to the number of large fractures. Figure 9(b) shows particle tortuosity and occupancy as a function of particle length from the exit plane, sampled at discrete points along particle traces. Most particles travel between 15 and 25 meters as they traverse the network; recall that minimum distance to traverse the cube is 12 meters. The tortuosity values indicate that most particles take a relatively direct path through the network. This is further emphasized by the graph in Figure 9(c), which relates fracture tortuosity and the topological distance of each fracture from the exit. The graph also shows that the majority of particles travel along seven fractures or less and that after each transition to a new fracture their path becomes more direct towards the exit.

5.2 Hydrocarbon extraction from unconventional resources

The process of hydraulic fracturing (fracking) involves injecting water at high pressures to create an interconnected network of fractures in an otherwise impermeably rock formation. This process increases the permeability of a low-permeability media, such as tight shales, allowing trapped hydrocarbons to be extracted. Using the methods of Karra et al., [24] we generate a DFN in a cubic kilometer domain based on surveys of a shale site in the upper Pottsville formation in Alabama; details of the site can be found in [39]. A horizontal well is included in the domain along with six equally spaced fractures that represent hydraulically generated fractures that are perpendicular to the well. Operational values for permeability, pressure, and porosity were used for realism. Particles representing packets of gas are seeded uniformly through out the DFN. Pumping at the well pulls particles from throughout the network to the well.

Figure 10(a) shows the hydraulic fractures, semi-transparent blue, the horizontal well, the rectangular plane in the center of the domain, and clusters of particle trajectories. The color of each trace corresponds to the cluster each belongs to. Figure 10(b) shows the backbones of the network, in white and grey, along with the hydraulic fractures,
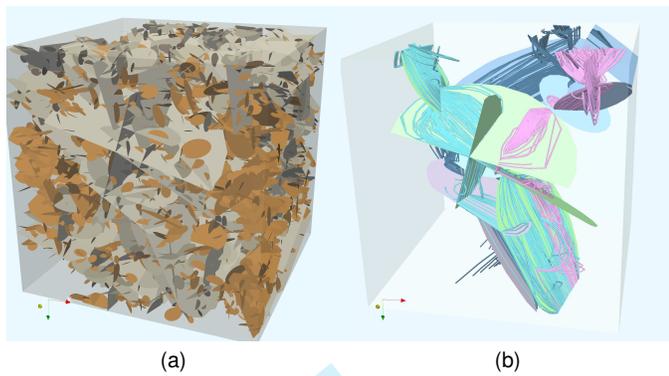
Fig. 11. (a) A DFN realization based on the fractured granite at the Forsmark site in Sweden. The domain is a cubic kilometer and contains approximately five thousand circular fractures whose radii are sampled from a truncated power-law distribution. The fractures highlighted orange indicate that these fracture have not been visited by any particle during the transport simulation. (b) The backbones of the DFN along with the three largest trace clusters colored by there cluster ID. The backbones are primarily comprised of larger fractures that act as conduits for flow and transport through the network, as shown by the clustering of traces.

shown in blue. To extract backbones for this network, we use a much lower percentage (0.5% as apposed to 3%) of exiting particles when considering the exit fractures. The reason such a small percentage is used for this DFN is that we wanted to capture any channelization that was occurring, but did not exit through one of the 6 large hydraulically generated fractures (circular and shown in blue for this figure). In this example there are several small backbone structures most of which, (all but four) connect to the hydraulically generated fractures before exiting through the well. These connections corroborate the empirical knowledge that fracking increases connectivity in the shale formation thereby making it easier for trapped hydrocarbons in low permeable shale to be extracted. Backbones and a representative trace from each cluster are plotted together in Figure 10(c). The thickness of each tube corresponds the number of particles in each cluster. The colors are consistent with (a). The representative traces primarily travel on the backbones indicating that clustering and flow channeling occurs on the backbones.

5.3 Kilometer DFN of fractured granite

The Swedish Nuclear Fuel and Waste Management Company (SKB) has undertaken a detailed investigation of the fractured granite at the Forsmark site, Sweden as a potential host formation for a subsurface repository for spent nuclear fuel [40]. We adopt a semi-generic subset of the statistical fracture model determined by SKB, details of the site characterization are provided in [40]. Our fracture model uses three fracture sets whose radii are determined by a truncated power-law distribution and varying orientations. The largest fractures have a radius of 560 meters and the smallest have a radius of 15 meters. An example network is shown in Figure 11 (a). The domain is a cubic kilometer and each realization contains approximately five thousand circular fractures. The fractures colored orange are not visited by a particle during the transport simulation; only 30%

of fractures in the domain are touched by a particle during transport simulations. These results indicate that the strong flow channeling is occurring along backbones in the DFN. Figure 11 (b) shows the backbones of the network along with the largest particle clusters for verification. The backbones are made up of large fractures and the particle trajectories tend to arrive on a fracture in a backbone and remain then remain along that path.

5.4 Network Comparison

DFN are generated stochastically and thus multiple realizations using the same underlaying statistics are required and these multiple DFN are compared to one another. This type of comparative analysis is desirable when trying to demonstrate ergodic behavior in upscaled transport distributions. For example, identifying universal fracture characteristics that lead to flow channeling, which is equivalent to particle clustering, requires numerous realizations. To demonstrate the utility of the proposed methodology in this regard, we compare networks generated using the same underlaying statistics. Comparisons between the networks are performed both visually and analytically to identify features and clusters in the networks.

Three independent DFN realizations based on the Forsmark site are created and the backbone of each network is determined, shown in Figure 12 (a-c). In the first and third realization, there is one large fracture that dominates transport through the system. In the second realization, shown in the middle, the backbone is made up of numerous medium sized fractures rather than a single large fracture. The methodology allows us to characterize and identify the key fracture characteristics that lead to flow channeling. One possible use of this methodology is to identify the characteristics of the fractures that make up the backbone and then create reduced DFN models that retain these backbones but omit fractures that do not significantly contribute to transport.

Figure 13 shows the particle tortuosity and percent occupancy of particles for the three DFN realizations. Although the backbones are different, particle ensemble statistics appear to have stabilized. Observed tortuosity values all scale linearly with path length to exit and little variability is observed between realizations. However, there are discrepancies in the observed cumulative distributions of percent of occupancy. Most notable, at large distances from the exit plane. One realization has fewer particles with long distances from the exit plane, and this is likely the result of the large fracture that dominants the backbone of DFN, cf. Figure 12 (a). In general, we can use such statistical comparisons to ensure that any given realization of the network topology is equally valid. If there are major discrepancies between networks, we can use the feature analysis and clustering to determine where these differences stem from.

6 CONCLUSIONS AND FUTURE WORK

We have introduced a flow topology graph method for the analysis of flow and transport in fractured rock that allows users to analyze simulated flow and transport in discrete fracture networks. Recent advancements in DFN
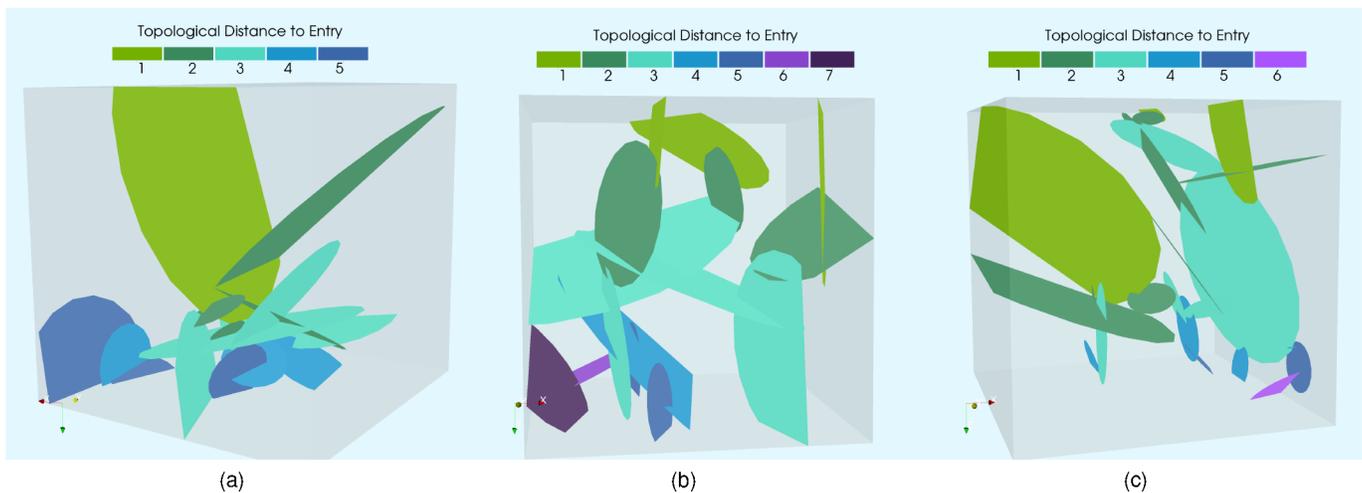
Fig. 12. The backbones in three realizations of the DFN network topology, all modeling the same physical domain of the Forsmark repository site. Each network has different types of main backbones. This type of comparison, between DFN modeling the same physical site, is important due to the stochastic nature in which DFN are produced.

(a)
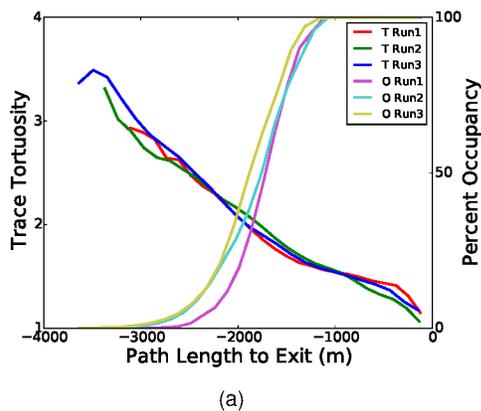
Fig. 13. Particle tortuosity and percent occupancy of particles for the DFN used in Figure 12, which are generated by sampling the same fracture statistics. Similarities and differences in the curves can be explained by the backbone structures developed in each realization.

simulation tools have made it possible to model and simulate flow at realistic scales with networks containing thousands of fractures. The methods presented here are part of a prototype system and toolset supporting the interactive, detailed exploration of simulated Lagrangian transport data. The methodology is both modular and flexible, allowing for rapid prototyping and modification of capabilities with changing goals and application needs. Visualization is decoupled from analysis, allowing users to interactively explore the results using tools they are familiar with.

Our FTG-based framework supports both global and localized statistical analysis, feature analysis for discovering channelization due to backbones, and intuitive clustering of particle paths in these large fracture networks. The methodology addresses the three main areas of analysis research identified by geoscientists concerned with flow and transport in fractured media: statistical analysis, topological path analysis, and topological trace clustering. The provided statistical analysis can be used to gain a better understanding of system-wide trends as well as identify potential problems in the simulation. The topological path analysis allows for the identification of important regions within the network, namely backbones, and allows for a systematic, integrative approach to identifying fracture characteristics that lead to flow channeling in fractured rock. The topological trace clustering identifies groups of particles that travel along similar paths and verify backbones. In combination, these tools can be used to identify geological structures that dictate flow and transport in the fractured rock. This characterization can potentially be utilized in the modeling of both static and adaptive control of subsurface processes, being relevant for areas including carbon sequestration, geothermal energy, contamination remediation, and unconventional oil and gas extraction.

While the results demonstrated with our prototype system are promising, it is important to note that more detailed case studies are needed to evaluate results more conclusively. More research needs to be done concerning the establishment of better metrics for improved path analysis, topological trace clustering, and comparative analysis. We have identified several metrics that produce reasonable results, and we currently allow a user to select what set of metrics to use. More studies are needed to determine which ones are most meaningful for a given DFN application. This aspect is especially relevant when performing comparative analysis, where a proper metric for measuring similarities and differences between multiple realizations of a DFN is crucially important (ensemble simulation and analysis). We do not currently directly visualize the graph as part of our analysis framework. However, developing a scalable algorithm for laying out the graph, especially for quickly comparing between multiple FTG, is the focus of ongoing research. Allowing users to directly interact with the FTG, and highlight features that may be difficult to describe algorithmically is also a future goal. Similarly, directly linking statistical plots with the geometry in a painting and linking style will be considered for future systems. This would require a more customized system, and thus some flexibility

in the visualizing tools would be lost, however the benefit may outweigh the cost. Currently our prototype system is used for performing data analysis in a post-processing step. Considering the increasing size and complexity of simulated DFNs, we will consider *in situ* use of our analyses. We have kept this goal in mind during the development of our prototype to minimize the amount of implementation that will need to be done when transitioning our system from a post-processing to an *in situ* analysis system.

REFERENCES

[1] R. S. Middleton, J. W. Carey, R. P. Currier, J. D. Hyman, Q. Kang, S. Karra, J. Jiménez-Martínez, M. L. Porter, and H. S. Viswanathan, "Shale gas and non-aqueous fracturing fluids: Opportunities and challenges for supercritical CO_2," *Appl. Energ.*, vol. 147, pp. 500–509, 2015.

[2] National Research Council (US). Committee on Fracture Characterization and Fluid Flow, *Rock fractures and fluid flow: contemporary understanding and applications*. National Academy Press, 1996.

[3] S. Neuman, "Trends, prospects and challenges in quantifying flow and transport through fractured rocks," *Hydrogeol. J.*, vol. 13, no. 1, pp. 124–147, 2005.

[4] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post, "Topology-Based Flow Visualization, The State of the Art," in *Topology-based Methods in Visualization*. Springer Berlin Heidelberg, 2007, pp. 1–19.

[5] T. Salzbrunn, H. Jänicke, and T. Wischgoll, "The State of the Art in Flow Visualization: Partition-Based Techniques." *SimVis*, pp. 75–92, 2008.

[6] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matković, and H. Hauser, "The State of the Art in Topology-Based Visualization of Unsteady Flow," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1789–1811, Sep. 2011.

[7] C. Jones and K.-L. Ma, "Visualizing flow trajectories using locality-based rendering and warped curve plots," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1587–1594, Nov 2010.

[8] T. Schultz, H. Theisel, and H.-P. Seidel, "Topological Visualization of Brain Diffusion MRI Data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1496–1503, 2007.

[9] B. Moberts, A. Vilanova, and J. van Wijk, "Evaluation of fiber clustering methods for diffusion tensor imaging," in *Visualization, 2005. VIS 05. IEEE*, Oct 2005, pp. 65–72.

[10] L. Xu and H.-W. Shen, "Flow Web: a graph based user interface for 3D flow field exploration," *IS&T/SPIE Electronic Imaging*, pp. 75 300F–75 300F–12, Jan. 2010.

[11] B. Nouanesengsy, T.-Y. Lee, and H.-W. Shen, "Load-Balanced Parallel Streamline Generation on Large Scale Vector Fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 1785–1794, Dec. 2011.

[12] C.-M. Chen and H.-W. Shen, "Graph-based seed scheduling for out-of-core FTLE and pathline computation," *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, pp. 15–23, 2013.

[13] G. Chen, K. Mischaikow, R. Laramee, P. Pilarczyk, and E. Zhang, "Vector field editing and periodic orbit extraction using morse decomposition," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 4, pp. 769–785, July 2007.

[14] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang, "Efficient Morse Decompositions of Vector Fields," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 4, pp. 848–862, 2008.

[15] G. Chen, Q. Deng, A. Szymczak, R. Laramee, and E. Zhang, "Morse set classification and hierarchical refinement using conley index," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 5, pp. 767–782, May 2012.

[16] T. McLoughlin, M. W. Jones, R. S. Laramee, R. Malki, I. Masters, and C. D. Hansen, "Similarity Measures for Enhancing Interactive Streamline Seeding," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 8, pp. 1342–1353, Apr. 2013.

[17] H. Yu, C. Wang, C.-K. Shene, and J. H. Chen, "Hierarchical Streamline Bundles," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 8, pp. 1353–1367, Aug. 2012.

[18] K. Lu, A. Chaudhuri, T.-Y. Lee, H.-W. Shen, and P. C. Wong, "Exploring vector fields with distribution-based streamline analysis." in *PacificVis*, 2013, pp. 257–264.

[19] J. Wei, H. Yu, R. W. Grout, J. H. Chen, and K.-L. Ma, "Dual space analysis of turbulent combustion particle data." *PacificVis*, pp. 91–98, 2011.

[20] J.-R. Dreuzy, Y. Méheust, and G. Pichot, "Influence of fracture scale heterogeneity on the flow properties of three-dimensional discrete fracture networks," *Journal of Geophysical Research - Solid Earth*, vol. 117, no. B11, 2012.

[21] J. Erhel, J.-R. De Dreuzy, and B. Poirriez, "Flow simulation in three-dimensional discrete fracture networks," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2688–2705, 2009.

[22] W. Dershowitz, "FracMan version 7.4-Interactive discrete feature data analysis, geometric modeling, and exploration simulation: User documentation," http://fracman.golder.com/, 2014.

[23] J. Hyman, S. Painter, H. Viswanathan, N. Makedonska, and S. Karra, "Influence of injection mode on transport properties in kilometer-scale three-dimensional discrete fracture networks," *Water Resources Research*, vol. 51, no. 9, pp. 7289–7308, 2015.

[24] S. Karra, N. Makedonska, H. S. Viswanathan, S. L. Painter, and J. D. Hyman, "Effect of advective flow in fractures and matrix diffusion on natural gas production," *Water Resources Research*, pp. n/a–n/a, 2015. [Online]. Available: http://dx.doi.org/10.1002/2014WR016829

[25] H. Mustapha and K. Mustapha, "A new approach to simulating flow in discrete fracture networks with an optimized mesh," *SIAM Journal of Scientific Computing*, vol. 29, p. 1439, 2007.

[26] J. D. Hyman, S. Karra, N. Makedonska, C. W. Gable, S. L. Painter, and H. S. Viswanathan, "dfnWorks : A discrete fracture network framework for modeling subsurface flow and transport," *Computers and Geosciences*, 2015, doi:10.1016/j.cageo.2015.08.001.

[27] J. D. Hyman, C. W. Gable, S. L. Painter, and N. Makedonska, "Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy," *SIAM J. Sci. Comput.*, vol. 36, no. 4, pp. A1871–A1894, 2014.

[28] LaGriT, "Los Alamos Grid Toolbox, (LaGriT) Los Alamos National Laboratory," http://lagrit.lanl.gov, 2013.

[29] P. Lichtner, G. Hammond, C. Lu, S. Karra, G. Bisht, B. Andre, R. Mills, and J. Kumar, "PFLOTRAN user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes," Los Alamos National Laboratory, Report No.: LA-UR-15-20403, Tech. Rep., 2015.

[30] S. L. Painter, C. W. Gable, and S. Kelkar, "Pathline tracing on fully unstructured control-volume grids," *Computat. Geosci.*, vol. 16, no. 4, pp. 1125–1134, 2012.

[31] N. Makedonska, S. Painter, Q. Bui, C. Gable, and S. Karra, "Particle tracking approach for transport in three-dimensional discrete fracture networks," *Computational Geosciences*, pp. 1–15, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10596-015-9525-4

[32] J.-R. de Dreuzy, P. Davy, and O. Bour, "Hydraulic properties of two-dimensional random fracture networks following power law

distributions of length and aperture," *Water Resour. Res.*, vol. 38, no. 12, 2002.

[33] J. D. Hyman, D. O'Malley, and J. Jiménez-Martínez, "Flow channeling, principal pathways, and correlated particle trajectories," *Water Resour. Res.*, 2016 (under review).

[34] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of Applied Mathematics*, vol. 27, no. 4, p. 526, 1970.

[35] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[36] R. Wilson and E. Hancock, "Levenshtein distance for graph spectral features," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, Aug 2004, pp. 489–492 Vol.2.

[37] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 1977. [Online]. Available: http://comjnl.oxfordjournals.org/content/20/4/364.abstract

[38] J. Ahrens, B. Geveci, and C. Law, "36 paraview: An end-user tool for large-data visualization," *The Visualization Handbook*, p. 717, 2005.

[39] G. Jin, J. Pashin, and J. Payton, "Application of discrete fracture network models to coalbed methane reservoirs of the Black Warrior basin: Tuscaloosa, Alabama, University of Alabama College of Continuing Studies," in *2003 International Coalbed Methane Symposium Proceedings, Paper*, vol. 321, 2003, p. 13.

[40] SKB, "Long-term safety for the final repository for spent nuclear fuel at forsmark. main report of the sr-site project." *SKB TR-11-01,*, 2011.

Carl Gable received his PhD in Geophysics from Harvard University in 1989. He is currently the group leader of the Computational Earth Science Group, Earth and Environmental Sciences Division, at Los Alamos National Laboratory. His research interest include unstructured finite element mesh generation, porous flow modeling, discrete fracture network modeling, and mantle convection.

Nataliia Makedonska received the PhD degree on applied mathematics and computer science from Weizmann Institute of Science in Israel. She is a scientist in Computational Earth Science Group at Los Alamos National Laboratory. Her research focuses on subsurface flow and transport modeling in three-dimensional discrete fracture networks with applications to contaminant transport studies and reservoir models.

Hari Viswanathan received his PhD degree in environmental engineering for the University of Illinois at Urbana-Champaign in 1999. He is the Subsurface Flow and Transport Team Leader in the Earth and Environmental Sciences Division at Los Alamos National Laboratory. He has over 70 publications in the area of subsurface energy. His research interests are flow and contaminant transport simulation for subsurface applications such as hydraulic fracturing, carbon sequestration and radioactive waste disposal.

Garrett Aldrich studied computer science and engineering at the University of California, Davis where he received his bachelors degree. He is currently a Computer Science PhD candidate at University of California, Davis, where his research interests include data analysis, feature extraction, scientific visualization, and parallel computing.

Jeffrey D. Hyman received his PhD in Applied Mathematics with a minor in Hydrology and Water Resources from the University of Arizona in 2014. He is currently a Director's Postdoctoral Fellow at Los Alamos National Laboratory. His research interest include detailed physical simulations of flow through large, kilometer-scale, fracture networks (discrete fracture networks) and small, micrometer-scale, explicit pore microstructures using high performance computing.

Jonathan Woodring is a research scientist at the Los Alamos National Laboratory. He received his PhD in computer science from The Ohio State University in 2009, specializing in computer graphics and scientific visualization. Jon's current research areas focus on data science at scale, scientific supercomputing, the intersection of high-performance and cloud computing, future power grids, and ocean climate modeling.

Satish Karra received his Phd in Mechanical Engineering from Texas A&M University in 2011.His research interests are modeling multiphysics/multiscale coupled processes in porous media, numerical methods for flow and transport in deforming porous media, and in developing high performance computing tools for these methods. His current research is being applied to LANL's efforts in subsurface applications such as Arctic hydrology, hydraulic fracturing, used fuel disposition, enhanced geothermal systems, carbon sequestration and contaminant transport.

Bernd Hamann studied computer science and mathematics at the Technical University of Braunschweig, Germany, and computer science at Arizona State University. At the University of California, Davis, his teaching and research areas are visualization, geometric modeling and computer graphics.