

**Predictions of first passage times in sparse discrete fracture networks using graph-based reductions**

Jeffrey D. Hyman\*

*Computational Earth Science (EES-16), Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Aric Hagberg and Gowri Srinivasan

*Applied Mathematics and Plasma Physics (T-5), Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Jamaludin Mohd-Yusof

*Applied Computer Science (CCS-7), Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

Hari Viswanathan

*Computational Earth Science (EES-16), Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

(Received 20 March 2017; published 10 July 2017)

We present a graph-based methodology to reduce the computational cost of obtaining first passage times through sparse fracture networks. We derive graph representations of generic three-dimensional discrete fracture networks (DFNs) using the DFN topology and flow boundary conditions. Subgraphs corresponding to the union of the  $k$  shortest paths between the inflow and outflow boundaries are identified and transport on their equivalent subnetworks is compared to transport through the full network. The number of paths included in the subgraphs is based on the scaling behavior of the number of edges in the graph with the number of shortest paths. First passage times through the subnetworks are in good agreement with those obtained in the full network, both for individual realizations and in distribution. Accurate estimates of first passage times are obtained with an order of magnitude reduction of CPU time and mesh size using the proposed method.

DOI: [10.1103/PhysRevE.96.013304](https://doi.org/10.1103/PhysRevE.96.013304)**I. INTRODUCTION**

Predicting the first passage time of solutes transported through a sparse fracture network is a common and critical challenge in many subsurface applications such as aquifer storage and management, environmental restoration of contaminated fractured media, the detection of low-level nuclear tests, CO<sub>2</sub> sequestration, and hydrocarbon extraction [1–8]. In low permeability media, interconnected fracture networks are the principal pathways for flow and the associated transport of dissolved solutes. In contrast to homogeneous porous media, heterogeneity resulting from the fracture networks creates irregular fluid velocity fields where flow channeling, isolated regions of high velocity, is commonly observed [9–14]. These flow channels indicate that there are subnetworks within the domain where the fastest transport occurs. In sparse fracture networks, the fractures that are included in these subnetworks are primarily determined by the macroscale structure of the network and the direction of flow [7,15] rather than mesoscale, e.g., fracture permeability [16,17], or microscale attributes, e.g., in-fracture aperture variability [12,18]. The *a priori* identification of these subnetworks using structural and hydrological properties would result in significant reductions in the computational demands for estimating first passage times because flow and transport simulations would only need to be performed in a subnetwork rather than the whole domain.

Discrete fracture network (DFN) models explicitly represent these structural and hydrological properties as discrete entities within an interconnected network of fractures. Individual fractures are  $(N - 1)$ -dimensional objects in an  $N$ -dimensional space, e.g., lines in two dimensions or planar polygons in three dimensions, and are assigned a shape, location, and orientation based on geological site characterizations. The fractures form an interconnected network embedded within an  $N$ -dimensional matrix that is considered impermeable. Once a network is constructed, the individual fractures are meshed for computation and the flow equations are numerically integrated to simulate flow and transport. The inclusion of detailed structural and hydrological properties allows DFN models to represent a wider range of transport phenomena than traditional continuum models [19,20]. In particular, topological, geometric, and hydrological characteristics can be directly linked to flow channeling and first passage times.

One limitation of DFN models is the high computational cost associated with the explicit representation of the fracture network. The number of mesh cells increases nonlinearly with the number of fractures, density of the network, and range of length scales being resolved. The computational overhead is particularly demanding if intersections between fractures are properly resolved in the mesh [21]. Because of limited computational resources, the first DFN models represented networks as a set of connected pipes [22] or used two-dimensional representations where the fractures did not need high resolution meshing [23]. Recent advances in high

\*Corresponding author: [jhyman@lanl.gov](mailto:jhyman@lanl.gov)

performance computing have allowed flow and transport simulations in large three-dimensional discrete fracture networks to be performed [24–30]. Nonetheless, the computational demands persist because fracture networks are stochastically generated due to uncertainty associated with their parameters. Therefore, numerous realizations are required to obtain stable statistics for upscaled observables such as first passage times.

We propose a graph-based method for efficient approximations of first passage times through sparse fracture networks. The cornerstone of the method is creating a graph representation of a DFN based on its topology and the flow boundary conditions. In sparse fracture networks whose radii exhibit a range of length scales, flow channel location (where the fastest transport through the network occurs) is primarily determined by the network structure [13,16,17] and the imposed flow direction [7]. Therefore, the proposed topologically based graph representation of the network captures one of the principal features that determines where flow channeling occurs. The other principal feature, the direction of flow, is incorporated by including the inflow and outflow boundaries into the graph. This construction allows us to identify subgraphs composed of the  $k$  shortest paths (fewest number of fractures) between the inflow and outflow boundaries. The mapping between the DFN and graph is a bijection so the preimage of each subgraph is a subnetwork that can be extracted from the full DFN. To obtain estimates of first passage times through the subnetwork, it is subject to the same meshing procedure, initial and boundary conditions for the pressure solution, and particle tracking as the full network.

While graphs and fracture networks have both been extensively studied, the application of graph theory to fracture network analysis is a fairly young discipline. Andresen *et al.* [31] proposed a similar transformation between DFNs and graphs to compare the topological structure of synthetic two-dimensional fracture networks with an actual rock outcrop. Hope *et al.* [32] used Andresen *et al.*'s representation to compare topological structures of two DFN generation methodologies in three dimensions. The focus of these studies was comparing fracture networks, rather than using the structure to identify subnetworks. Santiago *et al.* [33] constructed graphs based on two-dimensional fracture outcroppings and used topological measurements to identify the possible locations of flow channeling through the network. Aldrich *et al.* [34] introduced a weighted graph representation of three-dimensional fracture networks where edge weights were based on particle transport through the DFN. Analyzing this flow topology graph allowed them to identify the subnetwork where flow channeling occurred. Our method differs from that of Aldrich *et al.* [34] because we identify these subnetworks prior to running flow and transport simulations.

## II. DISCRETE FRACTURE NETWORK SIMULATIONS

### A. Generic fracture networks

We generated 100 three-dimensional generic fracture networks as a test set for the graph-reduction method. One hundred networks provide stable statistics in terms of transport breakthrough times (details not included). The networks

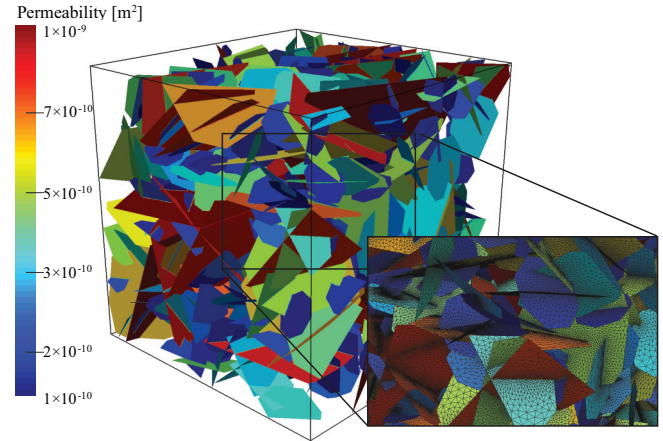


FIG. 1. Example discrete fracture network composed of 481 fractures. Fracture lengths are samples from a truncated power-law distribution and fracture permeabilities are positively correlated to the fracture radius. The inset shows the variable resolution conforming Delaunay triangulation of the fracture network.

are composed of circular fractures whose orientations are uniformly random and radii follow a truncated power-law distribution. Although the networks are meant to be generic, the network parameters are based on observed fractured media [35]. Each DFN is generated in a cubic domain with sides of length  $L = 15$  m. Fracture radii  $r$  (in meters) are sampled from a truncated power-law distribution with exponent  $\alpha$  and lower and upper cutoffs ( $r_0, r_u$ ),

$$r = r_0 \left[ 1 - u + u \left( \frac{r_0}{r_u} \right)^\alpha \right]^{-1/\alpha}, \quad (1)$$

where  $u$  is a random number sampled from the continuous uniform distribution on the closed interval  $[0, 1]$ . We select a value of  $\alpha = 2.6$  so that the distribution has finite mean and variance and is in accordance with geological observations [35]. The lower cutoff,  $r_0$ , is set to 1 m and the upper cutoff equal,  $r_u$ , is set to 5 m. Fracture centers are sampled uniformly throughout the domain. Isolated fractures and clusters that do not connect the inflow and outflow boundaries are removed because they do not contribute to flow. The resulting fracture networks contain around 500 fractures each.

Figure 1 shows a typical DFN realization. Fractures are colored by their permeabilities, with warmer colors indicating higher values. The inset shows the variable resolution conforming Delaunay triangulation of the fracture network generated using the feature rejection algorithm for meshing (FRAM) [21]. Mesh resolution is a function of distance from fracture intersections. The mesh is refined near fracture intersections to properly resolve the high gradients in the flow field that occur in these regions. The mesh is coarsened away from the intersections where gradients in the pressure field are smaller. Thus, the number of cells in the mesh increases with both the network surface area and density.

The fracture networks are sparse, with an average  $P_{32}$  value (total fracture surface area over total volume) of  $1.97 \text{ m}^{-1}$  and variance of 0.03. The connected network density [12] is about ten times the critical percolation value [36]. Thus,

the networks are dense enough that there are multiple paths between the inflow and outflow boundaries.

Variability in hydraulic properties is included into the network by correlating fracture apertures to their radii [14,17,37–41]. We use a positively correlated power-law relationship

$$b = \gamma r^\beta, \quad (2)$$

where  $\gamma = 5.0 \times 10^{-5}$  and  $\beta = 0.5$  are dimensionless parameters. It is possible to include in-fracture aperture variability into high fidelity DFN simulations [12,18], but constraining the in-fracture variability requires detailed knowledge of the particular rock formation. Therefore, we do not include in-fracture aperture variability in these simulations.

### B. Flow equations

Under this assumption of aperture uniformity, flow through the fractures is equivalent to flow between two parallel plates. The volumetric flow rate  $Q$  per unit fracture width normal to the direction of flow is therefore given by the Boussinesq equation [42],

$$Q = \frac{-b^3}{12\mu} \nabla P, \quad (3)$$

where  $\mu$  is the fluid viscosity and  $\nabla P$  is the pressure gradient. This relationship between aperture and flow rate can be used to derive a similar relationship between aperture and permeability,

$$k = \frac{b^2}{12}, \quad (4)$$

referred to as the cubic law [43]. A consequence of Eqs. (2) and (4) is that fracture permeability is positively correlated to its size.

Rewriting Eq. (3) using Eq. (4) provides the governing equation for flow within each two-dimensional fracture plane,

$$\mathbf{q} = -\frac{k}{\mu} \nabla P, \quad (5)$$

where  $\mathbf{q}$  is the Darcy flux ( $Q/b$ ), which is referred to as the Darcy equation.

We drive flow through the domain 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,

$$\nabla \cdot \mathbf{q} = 0, \quad (6)$$

and Eq. (5) are used to form an elliptic partial differential equation for steady-state distribution of pressure within each network:

$$\nabla \cdot (k \nabla P) = 0. \quad (7)$$

Once the distribution of pressure and volumetric flow rates are determined by numerically integrating Eq. (7), the methods of Makedonska *et al.* [44] and Painter *et al.* [45] are used to determine the Eulerian velocity field  $\mathbf{u}(\mathbf{x})$  within the DFN. Even though the fracture apertures are uniform within each fracture plane, the in-fracture velocity field is nonuniform. Variations in local flow fields depend on the local network

structure within each fracture plane. Specifically, intersections with other fractures influence the in-plane velocity field.

### C. Lagrangian attributes

We represent the spreading of a nonreactive conservative solute through each DFN by a cloud of passive tracer particles, i.e., using a Lagrangian approach. Complete mixing is used to determine what direction particles exit out of fracture intersections [44,46]. Particles do not interact with the matrix; i.e., matrix diffusion and sorption are not considered. The imposed pressure gradient is aligned with the  $x$  axis and thus the primary direction of flow is in the  $x$  direction. Particle initial positions  $\mathbf{a}$  are uniformly distributed along fracture intersections with the inlet plane  $\mathbf{x}_0 = (0, y, z)$ . The trajectory  $\mathbf{x}(t; \mathbf{a})$  of a particle starting at  $\mathbf{a}$  at time  $t = 0$  is given by the advection equation

$$\frac{d\mathbf{x}(t; \mathbf{a})}{dt} = \mathbf{v}(t; \mathbf{a}), \quad \mathbf{x}(0; \mathbf{a}) = \mathbf{a}, \quad (8)$$

where the Lagrangian velocity  $\mathbf{v}(t; \mathbf{x})$  is given in terms of the Eulerian velocity  $\mathbf{u}(\mathbf{x})$  as

$$\mathbf{v}(t; \mathbf{a}) = \mathbf{u}[\mathbf{x}(t; \mathbf{a})]. \quad (9)$$

The length of the path line,  $\ell$  (in meters), is used to parametrize the spatial and temporal coordinates of the particle. The space-time particle trajectory is given in terms of  $\ell$  by

$$\frac{d\mathbf{x}(\ell; \mathbf{a})}{d\ell} = \frac{\mathbf{v}[t(\ell); \mathbf{a}]}{v[t(\ell); \mathbf{a}]}, \quad (10a)$$

$$\frac{dt(\ell; \mathbf{a})}{d\ell} = \frac{1}{v[t(\ell), \mathbf{a}]}, \quad (10b)$$

where we set  $v(t, \mathbf{a}) = \|\mathbf{v}(t; \mathbf{a})\|$ . The length  $\ell(t; \mathbf{a})$  of the trajectory at a time  $t$  is

$$\frac{d\ell(t; \mathbf{a})}{dt} = v_\ell[\ell(t), \mathbf{a}], \quad (11)$$

where we defined  $v_\ell(\ell; \mathbf{a}) = v[t(\ell); \mathbf{a}]$ .

The travel time  $\tau(\mathbf{x}_L; \mathbf{a})$  of a particle that has crossed the outlet plane  $\mathbf{x}_L = (L, y, z)$  is

$$\tau(\mathbf{x}_L; \mathbf{a}) = t[\lambda(\mathbf{x}_L); \mathbf{a}], \quad (12)$$

where

$$\lambda(\mathbf{x}_L) = \inf\{\ell | x(\ell; \mathbf{a}) \geq L\}. \quad (13)$$

The first passage time of all particles through a network  $F$  is given by

$$\hat{\tau}(F) = \inf_{\{\mathbf{a}\} \in F} \{\tau(\mathbf{x}_L; \mathbf{a})\}. \quad (14)$$

We consider individual values of  $\hat{\tau}(F)$  and their distribution obtained for the ensemble of networks  $\Omega = \{F\}$ ,

$$\psi(t) = \int_{\Omega} dF \delta[t - \hat{\tau}(F)]. \quad (15)$$

We use the computational suite *dfnWorks* [27] to generate each three-dimensional DFN, solve the steady-state flow equations, and determine transport properties through the network. *dfnWorks* uses *FRAM* [21] to generate three-dimensional

fracture networks and the LaGriT meshing toolbox [47] to generate conforming Delaunay triangulation of the DFN. The parallelized subsurface flow and reactive transport code PFLOTRAN [48] is used to numerically integrate the governing flow equations to steady state. An extension of the WALKABOUT particle tracking method [44,45] is used to determine path lines through the DFN and simulate solute transport. Details of the suite, its abilities, applications, and references for detailed implementation are provided in [27].

### III. GRAPH REPRESENTATIONS

We construct a graph representation of each DFN based on the network topology. Let  $F = \{f_i\}$  for  $i = 1, \dots, n$  denote a DFN composed of  $n$  fractures. We define a mapping,  $\phi$ , 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  $u_i \in V$ ,

$$\phi : f_i \rightarrow u_i. \quad (16)$$

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,

$$\phi : f_i \cap f_j \neq \emptyset \rightarrow e_{ij} = (u_i, u_j), \quad (17)$$

where  $(u, v) \in E$  denotes an edge between vertices  $u$  and  $v$ . All edges are assigned unit edge weight to isolate topological attributes from other attributes that could be considered such as geometric, e.g., lengths, or hydrological, e.g., permeability. The mapping  $\phi$  is bijective; i.e., it is an isomorphism between  $F$  and  $G$ . Therefore, every subgraph  $G'(\{u\}, \{e\}) \subseteq G$  has a unique preimage  $F'$  in the fracture network,

$$\phi^{-1} : G' \rightarrow F', \quad (18)$$

that is a subnetwork of the full network,  $F' \subseteq F$ .

We 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,

$$\phi : f_i \cap \mathbf{x}_0 \neq \emptyset \rightarrow e_{si} = (s, u_i), \quad (19)$$

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

$$\phi : f_i \cap \mathbf{x}_L \neq \emptyset \rightarrow e_{it} = (u_i, t). \quad (20)$$

This mapping  $\phi$  is similar to the one proposed by Andresen *et al.* [31] but differs in this key aspect of including source and target vertices to represent inflow and outflow boundaries.

The considered fracture networks and mapping  $\phi$  results in graphs that have the following properties: (i) all vertices are degree one or greater because all fractures in the original network intersect at least one other fracture, and (ii) the graph is connected because each connected subnetwork within the DFN connects the inflow and outflow boundaries and is thus combined into the same graph via the source and target nodes; clusters that do not connect inflow and outflow boundaries do not contribute to flow and have been removed. A result of the second property is that there always exists at least one connected path between the source and target vertices.

Figure 2 shows the graph obtained from the fracture network shown in Fig. 1 using the mapping  $\phi$ . The source vertex is colored red and the target vertex is colored blue.

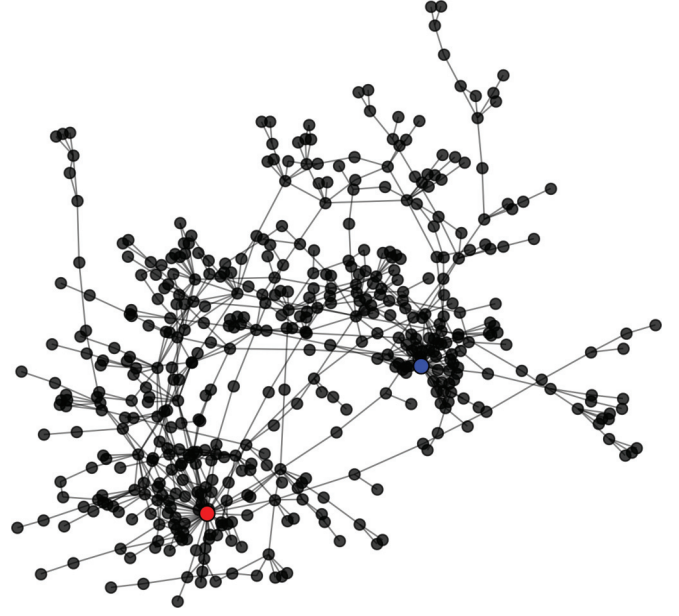


FIG. 2. Graph derived using the topology of the DFN shown in Fig. 1. A source vertex (red) has been included and connected to all fractures that intersect the inflow boundary and a target vertex (blue) has been included and connected at all fractures that intersect the outflow boundary.

The geometric layout of the graph is an arbitrary projection into  $\mathbb{R}^2$  using a force-directed layout algorithm [49]. Though the positions do not represent actual fracture locations in  $\mathbb{R}^3$  the drawing gives some perspective on the connectivity of the fracture network and the graph path lengths between the source and target.

#### Shortest-path subnetworks

We consider several subgraphs  $G'$ , along with their equivalent subnetworks  $F'$ , corresponding to the union  $U(k)$  of the edges in  $k$  shortest paths from the source to the target. The  $k$  shortest paths are defined as a generalization of the shortest path to include  $k$  total paths (possibly overlapping) in order of nondecreasing length starting from the shortest path. In our case we consider only loopless paths from the source to the target. The edges in  $G'$  have unit weight which we assign as the edge length; the shortest paths correspond to paths with the fewest edges between the source and the target. The preimage of this subgraph, which is its equivalent fracture subnetwork  $F'$ , has the fewest intersections, and thus connected fractures, spanning the inflow and outflow boundaries.

The number of shortest paths  $k$  to include in the subgraph  $U = U(k)$  is a parameter in the algorithm. To estimate a suitable value of  $k$  we calculated the shortest paths for various values of  $k$  and examined the resulting subgraph size. It is possible that for a given graph there are paths with the same length (in our case the same number of edges). Instead of optimizing the set of equal length paths to be included we instead increase the number of total paths,  $k$ , until we achieve the desired numerical result. Figure 3(a) shows the fraction of all edges  $|U|/|E|$  in the graph as a function of  $k$ . Thin semitransparent lines are individual network realizations and



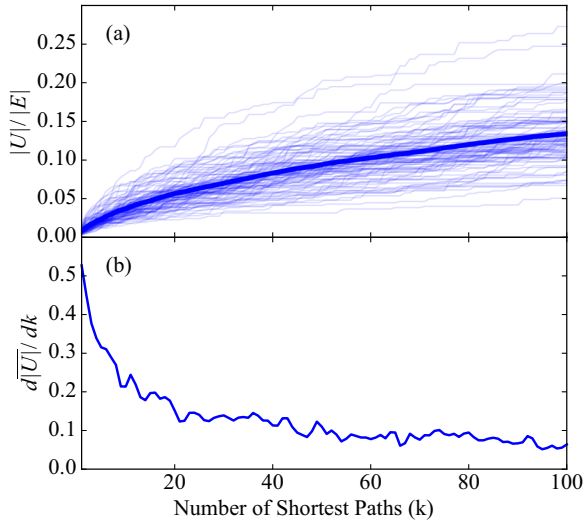


FIG. 3. (a) The fraction of edges  $|U|/|E|$  in the graph as a function of the number of shortest paths  $k$ . Thin semitransparent curves are individual network realizations and the thick curve is the average of all networks. (b) The rate of change for the average size  $\overline{|U|} = \sum |U|/100$  of the shortest-path edge set. The number of edges increases rapidly at first but after approximately  $k = 20$  the number of edges added with each new loopless path is small.

the thick line is the average  $\overline{|U|}$  of all 100 networks. Figure 3(b) shows the numerically estimated derivative of the average value  $\overline{|U|}$ . The average number of edges,  $\overline{|U|}$ , increases rapidly for small  $k$  but then for larger  $k$  few new edges are added with each additional path.

For first passage time calculations we start with the shortest paths between the source and target for a single path  $k = 1$ . Next, we consider the union of  $k$  shortest loopless paths from the source to target for  $k = 5, 10$ , and  $20$ . We select these values by considering the fraction of edges in the graph representation that are contained within each of these subgraphs. The values we select are in the fastest changing region (5 shortest paths), a moderate value (10 shortest paths), and the beginning of the region where the derivative has started to stabilize (20 shortest paths); cf. Fig. 3 (bottom).

We also consider the 2-core of the graph, which is an upper bound on the union of loopless paths from source to target. The  $k$ -core of a graph is the maximal subgraph that contains vertices of degree  $k$  or greater [50]. Physically, this set corresponds to fractures where transport can enter and exit a fracture through different intersections; e.g., all dead-end fractures are recursively removed. In three-dimensional fracture networks such dead-end fractures are not necessarily no-flow regions, which is the case in two dimensions. If the line of intersection between two fractures aligns with the pressure gradient there will be a gradient within the dead-end fractures and thus flow. Hence, the presence of dead-end fractures changes the local flow field on intersecting fractures and thus its removal does as well. The 2-core typically makes up between 50% and 60% of the graph edges (not shown in Fig. 3). The source and target vertices are always retained in the 2-core.

For a graph  $G$  with  $n$  vertices and  $m$  edges the shortest-path set can be computed using breadth-first search in  $O(m + n)$

time [51]. The computation of the  $k$  shortest paths is harder but still can be done in polynomial time,  $O(kn(m + n \log n))$  [52]. Computing the  $k$ -core composition has time complexity of  $O(m)$  [53]. The subgraphs sets are computed using the NETWORKX graph software package [54].

Figure 4 shows three subnetworks (top) and their subgraphs (bottom) derived from the network and graph shown in Figs. 1 and 2. Semitransparent vertices indicate fractures that have been eliminated from the fracture network. The full network is made of 481 fractures, the shortest path is made of 3 fractures (left), the ten shortest paths contain 23 fractures (middle), and the 2-core contains 276 fractures (right). This reduction in number of fractures drastically changes the number of cells in the mesh used for flow and transport simulations. The full network is meshed with 910 397 triangles, the shortest path is meshed with 5438 triangles, the ten shortest paths are meshed with 69 353 triangles, and the 2-core is meshed with 639 319 triangles.

The method to obtain first passage times using these subnetworks can be conceptually divided into the following four steps:

- (i) A graph representation of a DFN is constructed using the mapping defined in Eqs. (16) and (17) ( $\phi : F \rightarrow G$ ).
- (ii) A subgraph composed of the  $k$  shortest paths between the source and target is identified ( $G' \subseteq G$ ).
- (iii) We isolate the subnetwork that is the preimage of the extracted subgraph ( $\phi^{-1} : G' \rightarrow F'$ ).
- (iv) To obtain estimates of first passage times through each subnetwork, they are subject to the same meshing procedure, initial and boundary conditions for the pressure, and particle tracking initial conditions as the full network [Eqs. (7)–(14)].

#### IV. METHOD PERFORMANCE

We measure the method's performance in terms of accuracy and efficiency. First, we compare predictions of the first passage time in the full network and those obtained using each subnetwork. Second, we compare the computational cost for the simulations.

##### A. First passage times

Accuracy of the method is determined by comparing the first passage times (14) in  $F$  and  $F'$  and statistics of the distribution of first passage times (15) for the ensemble of networks  $\Omega = \{F\}$  and their subnetworks  $\Omega' = \{F'\}$ . Let  $\mathcal{L}$  denote the operator that takes a DFN  $F$  as an input and returns the first passage time  $\hat{\tau}$ ,

$$\mathcal{L} : F \rightarrow \hat{\tau}. \quad (21)$$

For each subnetwork  $F'$ , we can obtain a first passage time  $\hat{\tau}'$  using the same operator:

$$\mathcal{L} : F' \rightarrow \hat{\tau}'. \quad (22)$$

The goal is that the error

$$\|\hat{\tau} - \hat{\tau}'\| \quad (23)$$

is small for each realization and in terms of their distributions obtained from  $\Omega$  and  $\Omega'$ . Individual realizations provide a single value of  $\hat{\tau}'$  to directly compare with  $\hat{\tau}$ . The distributions of

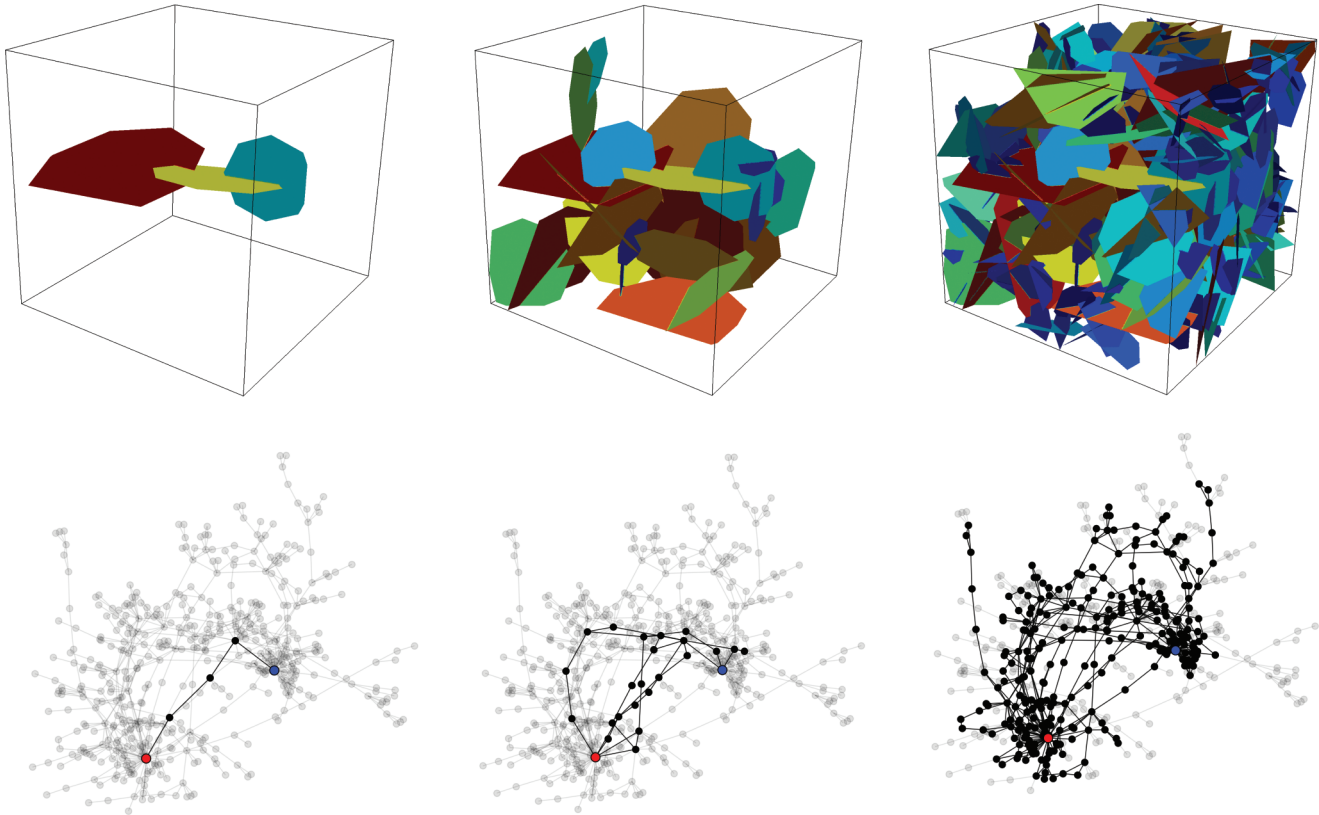


FIG. 4. Subnetworks (top) and subgraphs (bottom) derived from the network and graph shown in Figs. 1 and 2, respectively. Subfigures show the (left) shortest path through the network, (middle) union of the ten shortest paths in the network, and (right) 2-core network. Semitransparent vertices denote fractures that have been eliminated from the fracture network.

$\hat{\tau}'$  and  $\hat{\tau}$  are compared in terms of their first two moments. Differences between the distributions are measured by computing the Kullback-Leibler divergence (relative entropy), smaller values of which indicate better agreement between the two probability densities. The two-sample Kolmogorov-Smirnov test is also used to determine whether these differences are statistically significant.

Figures 5(a)–5(e) show  $\hat{\tau}'$  obtained in the subnetworks plotted against  $\hat{\tau}$ . The shortest path [Fig. 5(a)], 5 shortest paths [Fig. 5(b)], and 10 shortest paths [Fig. 5(c)] are on the top row and the 20 shortest paths [Fig. 5(d)] and 2-core [Fig. 5(e)] are on the bottom. Values are divided by the median passage time of the ensemble of particles through all 100 networks to nondimensionalize time. The black line corresponds to identical first passage times in the subnetworks and the full networks. Deviations are quantified by computing the coefficient of determination,  $R^2$ . Values of  $R^2$  that are closer to 1 indicate better agreement between  $\hat{\tau}'$  and  $\hat{\tau}$  over the set of sample networks (values are provided in Table I). In general, the first passage times of the shortest network (blue) are close to actual first passage times, but there are exceptions ( $R^2 = 0.54$ ). Values that deviate from the trend occur in networks where the particle with the earliest passage time does not travel along the shortest topological path. There is less scatter in the comparison with the five shortest paths ( $R^2 = 0.57$ ), but there are still outliers. Including the ten shortest paths leads to much better agreement with the full network ( $R^2 = 0.86$ ). However, the increase in accuracy by

increasing to include the 20 shortest paths is less than going from 5 to 10 ( $R^2 = 0.90$ ). The difference between the first passage times through the 2-core and the full networks is very small ( $R^2 = 0.99$ ).

Figure 5(f) shows the distributions of  $\hat{\tau}$  and  $\hat{\tau}'$  [Eq. (15)] obtained from the subnetworks and the full network. Table I reports the first two moments of the distributions (mean,  $\mu$ ; variance,  $\sigma^2$ ) of  $\hat{\tau}$  and  $\hat{\tau}'$  along with the results of the two-sample Kolmogorov-Smirnov (KS) test and Kullback-Leibler (KL) divergence measure. In general, there is good agreement between the distributions obtained using the subnetworks and the full network. The distribution of  $\hat{\tau}'$  in the shortest-path networks has a higher mean, a longer tail, and higher variance than the full network. As more paths are included into the subnetworks the KL divergence measure decreases, indicating better agreement with the distribution of first passage times in the full network. The 2-core matches the full network values well for all values. The two-sample Kolmogorov-Smirnov test rejects the null hypothesis that the distributions  $\hat{\tau}'$  from the shortest paths are from the same distribution of  $\hat{\tau}$  obtained for the full network. The first two moments of the distributions of  $\hat{\tau}'$  through the 5, 10, and 20 shortest paths and 2-core are close to those of  $\hat{\tau}$ . The two-sample Kolmogorov-Smirnov test returns low values of the KS statistic and high  $p$  values for the 5, 10, and 20 shortest paths. Comparison of the full network with the 2-core resulted in lower KS values and higher  $p$  values than any other subnetwork.

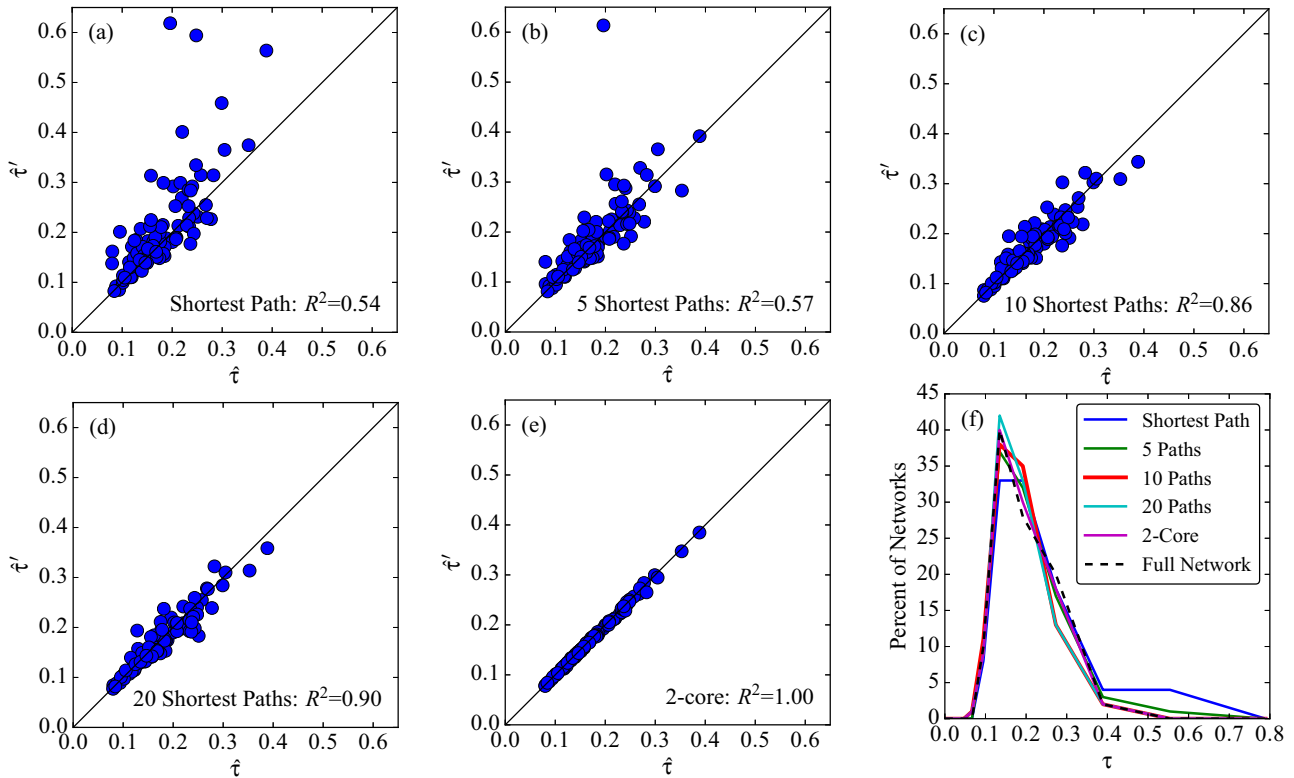


FIG. 5. First passage times in subnetworks vs the first passage time through the full network. Time has been nondimensionalized by the median breakthrough of the ensemble of particles through all 100 networks. The black line corresponds the identical first passage times in the subnetworks and the full networks. Values of coefficient of determination  $R^2$  closer to 1 indicate better agreement between  $\hat{\tau}'$  and  $\hat{\tau}$  over the set of sample networks. (a) Shortest path, (b) 5 shortest paths, (c) 10 shortest paths, (d) 20 shortest paths, and (e) 2-core. (f) The distributions of first passage times in all subnetworks and network realizations.

**B. Computational cost**

In this section we report mesh reduction and computational speed up when considering a subnetwork relative to the full network. Table II reports mean values of the total number of fractures, the total surface area of fractures, and the number of computational cells of the identified subnetworks along with respective percentages of the full networks for the DFN ensemble. On average, the shortest-path subnetworks make up  $\approx 2\%$  of the total number of fractures and  $\approx 7\%$  of the total surface area. These values indicate that the shortest paths are composed of a few large fractures that span the domain from the inlet to the outlet plane. The five shortest paths

TABLE I. Statistics of distributions for the first passage times through the network (mean,  $\mu$ ; variance,  $\sigma^2$ ), results of the two-sample Kolmogorov-Smirnov test, and the Kullback-Leibler (KL) divergence measure for subnetworks compared to the full network.

Subnetwork	$\mu$	$\sigma^2$	KS	$p$ value	KL
Shortest path	0.202	$9.73 \times 10^{-3}$	0.140	0.261	$6.35 \times 10^{-2}$
5 shortest paths	0.185	$6.02 \times 10^{-3}$	0.080	0.894	$3.01 \times 10^{-2}$
10 shortest paths	0.173	$3.19 \times 10^{-3}$	0.080	0.894	$1.59 \times 10^{-2}$
20 shortest paths	0.171	$3.21 \times 10^{-3}$	0.080	0.894	$6.99 \times 10^{-3}$
2-core	0.173	$3.51 \times 10^{-3}$	0.050	0.999	$1.03 \times 10^{-3}$
Full network	0.175	$3.61 \times 10^{-3}$			

contain a few more fractures and show an increase in surface area. On average, the ten shortest paths contain double the number of fractures as the shortest path but less than double the surface area. Because the shortest path is contained within the ten shortest paths, these values further indicate that the shortest paths are composed of larger fractures. The 2-core subnetworks contain  $\approx 56\%$  of the total number of fractures and  $\approx 75\%$  of the surface area. Thus, the complement of the 2-core, the dead-end fractures, makes up half of the network by number and consists of mostly small fractures.

The consequences of these reduced mesh sizes with respect to computational time are provided in Table III and shown in Fig. 6. The average required wall clock time for the three primary sections of the DFNWORKS workflow (network meshing, pressure simulation, and transport simulation) are provided. Computations are performed using a server that has 64 cores (1.4 GHz AMD Opteron™ processor 6272 with 2048 kB of cache each. Meshing is performed in parallel using either 16 cores or the total number of fractures, whichever is less. The flow solution is determined using 16 cores. Transport is performed using a single core. The time required for network generation prior to meshing is not included in the comparison because it is required for every network and subnetwork, but it is on the order of 1 s per network. The computation of the shortest paths,  $k$  shortest paths, and 2-core subgraphs using NETWORKX take less than 1 s each and those times are also omitted.

TABLE II. Mean values for geometric observables in the subnetworks identified using subgraphs: No. F, number of fractures; SA, network surface area; No. cells, number of computational cells. (·) are the percentages of these values when compared to the mean values of the total network.

Subnetwork	No. F	(%)	SA (m <sup>2</sup> )	(%)	No. cells	(%)
Shortest path	8.55	(1.91)	$6.12 \times 10^2$	(6.54)	$1.99 \times 10^4$	(2.44)
5 shortest paths	12.05	(2.69)	$8.12 \times 10^2$	(8.66)	$2.90 \times 10^4$	(3.56)
10 shortest paths	17.55	(3.92)	$1.08 \times 10^3$	(11.54)	$4.43 \times 10^4$	(5.44)
20 shortest paths	25.69	(5.74)	$1.43 \times 10^3$	(15.25)	$6.51 \times 10^4$	(8.07)
2-core	247.78	(55.39)	$7.07 \times 10^3$	(75.40)	$5.44 \times 10^5$	(66.88)
Full network	447.34	(100.00)	$9.37 \times 10^3$	(100.00)	$8.13 \times 10^5$	(100.00)

In terms of total run time, the  $k$ -shortest-paths subnetworks are over an order of magnitude faster than the full network. The networks contain significantly fewer fractures to mesh, which results in fewer degrees of freedom in the linear system of pressure and faster solver convergence. The time required for transport is also drastically reduced because a smaller number of fractures intersect the inlet plane and thus fewer particles are inserted into the domain. The time required for the 2-core is less than that for the full network, but is the same order of magnitude.

## V. REMARKS

We have presented a graph-based method to reduce the computational cost of obtaining first passage times through sparse fracture networks. The graph representation of the DFN is derived using the network topology and flow boundary conditions. The preimage of each subgraph is a unique subnetwork because the mapping between the DFN and the graph is a bijection. All edges in the graphs have unit weight, so the shortest topological paths in the graph, which have the fewest edges between the source and the target, correspond to the fewest fractures between the inflow and outflow boundaries in the DFN. The subnetworks corresponding to the shortest topological paths tend to be composed of large fractures that are the principal highways for transport through the network (Table II). Once the primary paths have been identified, the size of the fractures added to the subnetworks decreases with additional numbers of shortest paths included into the subgraphs. This stabilization of the subnetwork structure is why the number of edges in the subgraphs plateaus as the number of shortest paths increases; its derivative decreases rapidly and then tends towards zero (Fig. 3).

In scientific computing there is commonly a tradeoff between accuracy and efficiency. Here, the tradeoff is clear

when considering the values reported in Tables I, II, and III and images shown in Figs. 5 and 6 that compare first passage times predicted using subnetworks and their associated computational cost. While the shortest path and the five shortest paths require the smallest CPU times, they provide the worst estimates of first passage times. This inaccuracy is apparent from the wide scatter seen in Fig. 5 that is quantified by the low values of the coefficient of determination ( $R^2 = 0.54$  and  $R^2 = 0.57$ ). Using the ten shortest paths requires slightly more CPU time, but the predictions of first passage times are significantly improved ( $R^2 = 0.86$ ). The primary paths through the network, discussed above, are included in the first ten shortest paths for all networks. Thus, the 20 shortest paths resulted in only minor modifications to the subnetworks and relatively little increase in accuracy ( $R^2 = 0.90$ ). The 2-core of the graph, an upper bound on all shortest paths between the source and target, provided the best predictions of first passage times ( $R^2 = 0.99$ ). However, the CPU time required for computation on the 2-core subnetwork was 75% of that needed for the full network, underscoring the aforementioned tradeoff between accuracy and efficiency. In terms of the distributions of first passage times from the entire set of networks, values obtained in the 10- and 20-shortest-paths networks and the 2-core were very similar to those obtained from the full networks [Fig. 5(f) and Table I].

The variable mesh resolution with local refinement around intersections plays a subtle role in reducing mesh size and the cost of computing first passage times. The number of cells in the mesh is proportional to the fracture surface area and the density of the subnetwork, which is significantly less for the subnetworks compared to the full network. When fractures are excluded from a subnetwork, their intersections on retained fractures do not exist. Thus, there are fewer intersections in the subnetworks that are refined with a high resolution conforming

TABLE III. Mean in wall clock time (seconds) for meshing, flow, and transport simulations in the full networks and subnetworks. Meshing is performed with either 16 cores or the total number of fractures, whichever is less. Flow solutions are performed using 16 cores. Transport is performed using a single core. (·) are the percentages of these values when compared to the mean values of the total network.

Network	Meshing (s)	(%)	Flow (s)	(%)	Transport (s)	(%)	Total (s)	(%)
Shortest path	21.99	(5.28)	7.65	(4.27)	19.67	(1.88)	49.31	(3.00)
5 shortest paths	38.20	(9.17)	10.69	(5.96)	24.57	(2.35)	73.46	(4.47)
10 shortest paths	39.35	(9.44)	8.17	(4.56)	32.60	(3.11)	80.12	(4.87)
20 shortest paths	64.92	(15.58)	15.47	(8.63)	57.78	(5.51)	138.17	(8.41)
2-core	287.69	(68.35)	102.79	(57.30)	874.64	(83.47)	1265.12	(76.96)
Full network	416.68	(100.00)	179.36	(100.00)	1047.75	(100.00)	1643.79	(100.00)



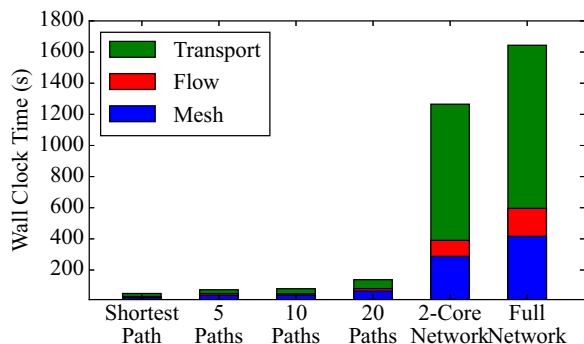


FIG. 6. Average wall clock time required for meshing (blue), flow simulation (red), and transport (green). Values are provided in Table III. In total run time,  $k$ -shortest-path subnetworks are over an order of magnitude faster than the full network. The time required for the 2-core, an upper bound on all shortest paths from source to target, is less than that for the full network, but is the same order of magnitude.

mesh and the number of cells in the mesh is reduced by more than the number of cells on the omitted fractures.

First passage times through the subnetworks deviate from those obtained using the full network for a number of reasons. The largest differences occur in networks where the fastest particle does not travel along the shortest topological path. In such situations, the subnetwork based on the shortest path cannot produce the desired value. The slowest passage times through the shortest-path subnetworks relative to the full network are the result of this issue [cf. top left corner of Fig. 5(a)]. Including the five shortest paths mostly alleviates this problem, but there are still networks where the fastest path is not contained in these subnetworks. The union of the ten shortest paths provides much better agreement with the full network because the path taken by the fastest particle is always contained within the first ten shortest paths. Even here, however, the match is not perfect. In this case, deviations in first passage times are due to differences between the in-fracture flow fields in the subnetwork and the full network. When fractures are omitted from a subnetwork, the in-fracture velocity field is different from the full network because it is sensitive to the in-plane geometry, e.g., the intersections with other fractures. The absence of these intersections decreases in-plane dispersion and can reduce travel time. For the 2-core subnetworks, the in-plane flow fields are more similar to the full networks because fewer fractures have been omitted. The absence of dead-end fractures modifies the local flow field on the remaining fracture planes because dead-end fractures are not necessarily no-flow regions, as is the case in two-dimensional simulations.

How many paths are needed to obtain good approximations for the fastest travel times is linked to how much of the graph is included (Fig. 3) and will vary with different DFN generation parameters. When the amount of the graph included with additional shortest paths is changing rapidly, the predicted values of first passage times are less accurate than when the derivative of this function is relatively stable. For the networks we considered, the ten shortest paths are a reasonable choice because it balanced accuracy and computational efficiency. For different network structures one should examine the scaling of edge counts with the number of shortest paths to select an appropriate number of paths. A conservative estimate for the number of shortest paths needed would be the value of  $k$  where this function's derivative has flattened out [Fig. 3(b)]. However, if this function does not stabilize, then the proposed method will likely not perform well.

Our test DFN set is composed of sparse semigenetic fracture networks whose radii follow a truncated power-law distribution, similar to many observed fracture sites [35]. Under the assumption of a positive correlation between a fracture's size and its transmissivity, the large fractures that make up the shortest paths are both the principal geometric pathways and hydrological fast paths. Hyman *et al.* [17] found that the adoption of this correlation did not significantly influence where the majority of transport occurred in similar networks, only how fast it traveled. Therefore, the proposed method should work in sparse networks where this correlation is weaker or even nonexistent, but only if the principal constraints on flow field structure are topological rather than hydrological. In sparse fracture networks, flow structure is primarily determined by the macroscale structure of the network and the direction of flow. In dense networks, mesoscale features, e.g., fracture permeability, and microscale attributes, e.g., in-fracture aperture variability, might be the principal controls of the flow structure. In these networks, the incorporation of hydrological parameters into the graph representation might be required to properly identify the subnetworks where flow channeling occurs. Such incorporations warrant further investigations and are the subject of ongoing research.

#### ACKNOWLEDGMENTS

This work was supported by the Laboratory Directed Research and Development program at Los Alamos National Laboratory under Department of Energy Contract No. DE-AC52-06NA25396. J.D.H. thanks the LANL LDRD Director's Postdoctoral Fellowship Grant No. 20150763PRD4 for partial support. All the authors thank LANL LDRD-DR Grant No. 20170103DR for support. Los Alamos unlimited release number 17-22022.

[1] J. Hyman, J. Jiménez-Martínez, H. Viswanathan, J. Carey, M. Porter, E. Rougier, S. Karra, Q. Kang, L. Frash, L. Chen *et al.*, *Philos. Trans. R. Soc. A* **374**, 20150426 (2016).

[2] C. Jenkins, A. Chadwick, and S. D. Hovorka, *Int. J. Greenhouse Gas Control* **40**, 312 (2015).

[3] S. Karra, N. Makedonska, H. Viswanathan, S. Painter, and J. Hyman, *Water Resour. Res.* **51**, 8646 (2015).

- [4] B. H. Kueper and D. B. McWhorter, *Ground Water* **29**, 716 (1991).
- [5] R. Middleton, J. Carey, R. Currier, J. Hyman, Q. Kang, S. Karra, J. Jiménez-Martínez, M. Porter, and H. Viswanathan, *Appl. Energy* **147**, 500 (2015).
- [6] National Research Council, *Rock Fractures and Fluid Flow: Contemporary Understanding and Applications* (National Academy Press, Washington, DC, 1996).
- [7] S. Neuman, *Hydrogeol. J.* **13**, 124 (2005).
- [8] J. VanderKwaak and E. Sudicky, *J. Contam. Hydrol.* **23**, 45 (1996).
- [9] H. Abelin, I. Neretnieks, S. Tunbrant, and L. Moreno, Nationale Genossenschaft für die Lagerung Radioaktiver Abfälle report, 1985 (unpublished).
- [10] H. Abelin, L. Birgersson, L. Moreno, H. Widén, T. Ågren, and I. Neretnieks, *Water Resour. Res.* **27**, 3119 (1991).
- [11] A. Rasmuson and I. Neretnieks, *Water Resour. Res.* **22**, 1247 (1986).
- [12] J.-R. de Dreuzy, Y. Méheust, and G. Pichot, *J. Geophys. Res.* **117**, B11207 (2012).
- [13] A. Frampton and V. Cvetkovic, *Water Resour. Res.* **47**, 1 (2011).
- [14] J. D. Hyman, S. L. Painter, H. Viswanathan, N. Makedonska, and S. Karra, *Water Resour. Res.* **51**, 7289 (2015).
- [15] P. Grindrod and M. Impey, *Water Resour. Res.* **29**, 4077 (1993).
- [16] J.-R. de Dreuzy, P. Davy, and O. Bour, *Water Resour. Res.* **37**, 2079 (2001).
- [17] J. D. Hyman, G. Aldrich, H. Viswanathan, N. Makedonska, and S. Karra, *Water Resour. Res.* **52**, 6472 (2016).
- [18] N. Makedonska, J. D. Hyman, S. Karra, S. L. Painter, C. W. Gable, and H. S. Viswanathan, *Adv. Water Resour.* **94**, 486 (2016).
- [19] S. Painter and V. Cvetkovic, *Water Resour. Res.* **41**, W02002 (2005).
- [20] S. Painter, V. Cvetkovic, and J.-O. Selroos, *Geophys. Res. Lett.* **29**, 20-1 (2002).
- [21] J. D. Hyman, C. W. Gable, S. L. Painter, and N. Makedonska, *SIAM J. Sci. Comput.* **36**, A1871 (2014).
- [22] M. C. Cacas, E. Ledoux, G. De Marsily, A. Barbreau, P. Calmels, B. Gaillard, and R. Margritta, *Water Resour. Res.* **26**, 491 (1990).
- [23] J.-R. de Dreuzy, C. Darcel, P. Davy, and O. Bour, *Water Resour. Res.* **40**, W01502 (2004).
- [24] S. Berrone, S. Pieraccini, and S. Scialo, *SIAM J. Sci. Comput.* **35**, B487 (2013).
- [25] S. Berrone, S. Pieraccini, S. Scialò, and F. Vicini, *SIAM J. Sci. Comput.* **37**, C285 (2015).
- [26] J. Erhel, J.-R. de Dreuzy, and B. Poirriez, *SIAM J. Sci. Comput.* **31**, 2688 (2009).
- [27] J. D. Hyman, S. Karra, N. Makedonska, C. W. Gable, S. L. Painter, and H. S. Viswanathan, *Comput. Geosci.* **84**, 10 (2015).
- [28] H. Mustapha and K. Mustapha, *SIAM J. Sci. Comput.* **29**, 1439 (2007).
- [29] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *Appl. Anal.* **89**, 1629 (2010).
- [30] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *SIAM J. Sci. Comput.* **34**, B86 (2012).
- [31] C. A. Andresen, A. Hansen, R. Le Goc, P. Davy, and S. M. Hope, *Front. Phys.* **1**, 7 (2013).
- [32] S. M. Hope, P. Davy, J. Maillot, R. Le Goc, and A. Hansen, *Front. Phys.* **3**, 75 (2015).
- [33] E. Santiago, J. X. Velasco-Hernández, and M. Romero-Salcedo, *Expert Syst. Appl.* **41**, 811 (2014).
- [34] G. Aldrich, J. Hyman, S. Karra, C. Gable, N. Makedonska, H. Viswanathan, J. Woodring, and B. Hamann, *IEEE Trans. Visualization Comput. Graphics* **23**, 1896 (2017).
- [35] E. Bonnet, O. Bour, N. E. Odling, P. Davy, I. Main, P. Cowie, and B. Berkowitz, *Rev. Geophys.* **39**, 347 (2001).
- [36] O. Bour and P. Davy, *Water Resour. Res.* **33**, 1567 (1997).
- [37] I. I. Bogdanov, V. V. Mourzenko, J.-F. Thovert, and P. M. Adler, *Phys. Rev. E* **76**, 036309 (2007).
- [38] J.-R. de Dreuzy, P. Davy, and O. Bour, *Water Resour. Res.* **38**, 12-1 (2002).
- [39] A. Frampton and V. Cvetkovic, *Water Resour. Res.* **46**, 1 (2010).
- [40] S. Joyce, L. Hartley, D. Applegate, J. Hoek, and P. Jackson, *Hydrogeol. J.* **22**, 1233 (2014).
- [41] T. P. Wellman, A. M. Shapiro, and M. C. Hill, *Water Resour. Res.* **45**, 1 (2009).
- [42] J. Boussinesq, *J. Math. Pure Appl.* **13**, 377 (1868).
- [43] P. A. Witherspoon, J. Wang, K. Iwai, and J. Gale, *Water Resour. Res.* **16**, 1016 (1980).
- [44] N. Makedonska, S. L. Painter, Q. M. Bui, C. W. Gable, and S. Karra, *Comput. Geosci.* **19**, 1123 (2015).
- [45] S. L. Painter, C. W. Gable, and S. Kelkar, *Comput. Geosci.* **16**, 1125 (2012).
- [46] P. K. Kang, M. Dentz, T. Le Borgne, and R. Juanes, *Phys. Rev. E* **92**, 022148 (2015).
- [47] LaGriT, Los Alamos Grid Toolbox (Los Alamos National Laboratory, 2013), <http://lagrit.lanl.gov>.
- [48] P. Lichtner, G. Hammond, C. Lu, S. Karra, G. Bisht, B. Andre, R. Mills, and J. Kumar, Los Alamos National Laboratory Report No. LA-UR-15-20403, 2015 (unpublished).
- [49] T. M. J. Fruchterman and E. M. Reingold, *Software: Pract. Exper.* **21**, 1129 (1991).
- [50] S. B. Seidman, *Soc. Networks* **5**, 269 (1983).
- [51] R. Sedgewick and K. Wayne, *Algorithms* (Addison-Wesley, Reading, MA, 2016).
- [52] J. Yen, *Manage. Sci.* **17**, 712 (1971).
- [53] V. Batagelj and M. Zaversnik, [arXiv:cs/0310049](https://arxiv.org/abs/cs/0310049).
- [54] A. A. Hagberg, D. A. Schult, and P. Swart, in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, Vol. 2008 (SciPy Organizers, Pasadena, CA, 2008), pp. 11–16.