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

Jeffrey D. Hyman, Aric Hagberg, Gowri Srinivasan, Jamaludin Mohd-Yusof, and Hari Viswanathan

Accurate and efficient 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, USA

Aric Hagberg and Gowri Srinivasan

Applied Mathematics and Plasma Physics (T-5), Theoretical Division,

Los Alamos National Laboratory, Los Alamos New Mexico, USA

Jamaludin Mohd-Yusof

Applied Computer Science (CCS-7), Computer,

Computational, and Statistical Sciences Division,

Los Alamos National Laboratory, Los Alamos New Mexico, USA

Hari Viswanathan

Computational Earth Science (EES-16),

Earth and Environmental Sciences Division,

Los Alamos National Laboratory, Los Alamos New Mexico, USA

Abstract

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 (DFN) using the DFN topology and flow boundary conditions. Subgraphs corresponding to the union of the *k* shortest paths between the inflow to outflow boundaries are identified and transport on their equivalent subnetworks is compared to transport through the full network. The number of paths included 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.

* Corresponding Author e-mail:jhyman@lanl.gov

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_2 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 macro-scale structure of the network and the direction of flow [7, 15] rather than meso-scale, e.g., fracture permeability [16, 17] or micro-scale 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 networks (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

50 overhead is particularly demanding if intersections between fractures are properly resolved in the

51 mesh [21]. Because of limited computational resources, the first DFN models represented net-

52 works as a set of connected pipes [22] or used two-dimensional representations where the frac-

53 tures did not need high resolution meshing [23]. Recent advances in high performance computing

54 have allowed flow and transport simulations in large three-dimensional discrete fracture networks

55 to be performed [24–30]. Nonetheless, the computational demands persist because fracture net-

56 works are stochastically generated due to uncertainty associated with their parameters. Therefore,

57 numerous realizations are required to obtain stable statistics for upscaled observables such as first

58 passage times.

59   We propose a graph-based method for efficient approximations of first passage times through

60 sparse fracture networks. The cornerstone of the method is creating a graph representation of a

61 DFN based on its topology and the flow boundary conditions. In sparse fracture networks whose

62 radii exhibit a range of length scales, flow channel location (where the fastest transport through

63 the network occurs) is primarily determined by the network structure [13, 16, 17] and the imposed

64 flow direction [7]. Therefore the proposed topologically based graph representation of the net-

65 work captures one of the principal features that determines where flow channeling occurs. The

66 other principal feature, the direction of flow, is incorporated by including the inflow and outflow

67 boundaries into the graph. This construction allows us to identify subgraphs composed of the k

68 shortest paths (fewest number of fractures) between the inflow and outflow boundaries. The map-

69 ping between the DFN and graph is a bijection so the pre-image of each subgraph is a subnetwork

70 that can be extracted from the full DFN. To obtain estimates of first passage times through the

71 subnetwork, it is subject to the same meshing procedure, initial and boundary conditions for the

72 pressure solution, and particle tracking, as the full network.

73   While graphs and fracture networks have both been extensively studied, the application of

74 graph theory to fracture network analysis is a fairly young discipline. Andresen et al. [31] pro-

75 posed a similar transformation between DFNs and graphs to compare the topological structure of

76 synthetic two-dimensional fracture networks with an actual rock outcrop. Hope et al. [32] used

77 Andresen's representation to compare topological structures of two DFN generation methodolo-

78 gies in three-dimensions. The focus of these studies was comparing fracture networks, rather

79 than using the structure to identify subnetworks. Santiago et al. [33] constructed graphs based on

80 two-dimensional fracture outcroppings and used topological measurements to identify the possible

81 locations of flow channeling through the network. Aldrich et al. [34] introduced a weighted graph

82 representation of three-dimensional fracture networks where edge weights were based on particle

83 transport through the DFN. Analyzing this flow topology graph allowed them to identify the sub-

84 network where flow channeling occurred. Our method differs from Aldrich et al. [34] because we

85 identify these subnetworks prior to running flow and transport simulations.

II. DISCRETE FRACTURE NETWORK SIMULATIONS

A. Generic Fracture Networks

88 We generated 100 three-dimensional generic fracture networks as a test set for the graph-

89 reduction method. One hundred networks provide stable statistics in terms of transport break-

90 through times (details not included). The networks are composed of circular fractures whose

91 orientations are uniformly random and radii follow a truncated power law distribution. Although

92 the networks are meant to be generic, the network parameters are based on observed fractured me-

93 dia [35]. Each DFN is generated in a cubic domain with sides of length $L = 15$ meters. Fracture

94 radii r [m] are sampled from a truncated power law distribution with exponent α and lower and

95 upper cutoffs $(r_0; r_u)$,

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

96 where u is a random number sampled from the continuous uniform distribution on the closed

97 interval [0,1]. We select a value of $\alpha = 2.6$ so that the distribution has finite mean and variance

98 and is in accordance with geological observations [35]. The lower cut off, r_0, is set to one meter

99 and the upper cut off equal, r_u, is set to five meters. Fracture centers are sampled uniformly

100 throughout the domain. Isolated fractures and clusters that do not connect the inflow and outflow

101 boundaries are removed because they do not contribute to flow. The resulting fracture networks

102 contain around 500 fractures each.

103 Figure 1 shows a typical DFN realization. Fractures are colored by their permeabilities with

104 warmer colors indicating higher values. The inset shows the variable resolution conforming De-

105 launay triangulation of the fracture network generated using the feature rejection algorithm for

106 meshing (FRAM) [21]. Mesh resolution is a function of distance from fracture intersections. The

107 mesh is refined near fracture intersections to properly resolve the high gradients in the flow field

108 that occur in these regions. The mesh is coarsened away from the intersections where gradients

109 in the pressure field are smaller. Thus, the number of cells in the mesh increases with both the
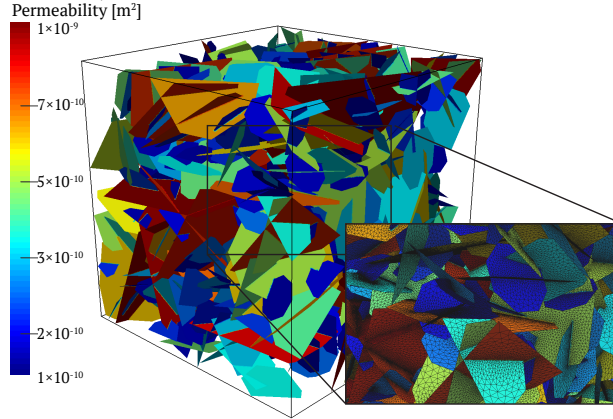
5

110 network surface area and density.

Figure 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. 111 The inset shows the variable resolution conforming Delaunay triangulation of the fracture network.

112

113   The fracture networks are sparse, with an average P_{32} value (total fracture surface area over 114 total volume) of 1.97 [m^{-1}] and variance of 0.03. The connected network density [12] is about 115 ten times the critical percolation value [36]. Thus, the networks are dense enough that there are 116 multiple paths between the inflow and outflow boundaries.

117   Variability in hydraulic properties is included into the network by correlating fracture apertures 118 to their radii [14, 17, 37–41]. We use a positively correlated power-law relationship

$$b = \gamma r^{\beta}, \tag{2}$$

119 where $\gamma = 5.0 \times 10^{-5}$ and $\beta = 0.5$ are dimensionless parameters. It is possible to include in- 120 fracture aperture variability into high fidelity DFN simulations [12, 18], but constraining the in- 121 fracture variability requires detailed knowledge of the particular rock formation. Therefore, we do 122 not include in-fracture aperture variability in these simulations.

123   **B.   Flow Equations**

124   Under this assumption of aperture uniformity, flow through the fractures is equivalent to flow 125 between two parallel plates. The volumetric flow rate Q per unit fracture width normal to the 126 direction of flow is therefore given by the Boussinesq equation [42],

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

127 where μ is the fluid viscosity, and ∇P it the pressure gradient. This relationship between aperture
128 and flow rate can be used to derive a similar relationship between aperture and permeability

$$k = \frac{b^2}{12}, \tag{4}$$

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

131 Rewriting (3) using (4) provides the governing equation for flow within each two dimensional
132 fracture plane,

$$\mathbf{q} = -\frac{k}{\mu}\nabla P, \tag{5}$$

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

134 We drive flow through the domain by applying a pressure difference of 1MPa across the do-
135 main aligned with the x-axis. No flow boundary conditions are applied along lateral boundaries
136 and gravity is not included in these simulations. These boundary conditions along with mass
137 conservation,

$$\nabla \cdot \mathbf{q} = 0, \tag{6}$$

138 and equation (5) are used to form an elliptic partial differential equation for steady-state distribu-
139 tion of pressure within each network

$$\nabla \cdot (k \nabla P) = 0. \tag{7}$$

140 Once the distribution of pressure and volumetric flow rates are determined by numerically inte-
141 grating (7), the methods of Makedonska et al. [44] and Painter et al. [45] are used to determine
142 the Eulerian velocity field $\mathbf{u}(\mathbf{x})$ within the DFN. Even though the fracture apertures are uniform
143 within each fracture plane, the in-fracture velocity field is non-uniform. Variations in local flow
144 fields depend on the local network structure within each fracture plane. Specifically, intersections
145 with other fractures influence the in-plane velocity field.

146 **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 **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 **a** at time $t = 0$ is given by the advection equation

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

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

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

The length of the pathline [m], ℓ, is used to parameterize the spatial and temporal coordinates of the particle. The space-time particle trajectory is given in terms of ℓ by

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

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

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

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

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

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

where

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

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

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

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

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

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.

155 DFNWORKS uses the feature rejection algorithm for meshing (FRAM) [21] to generate three-
156 dimensional fracture networks and the LaGriT meshing toolbox [47] to generate conforming De-
157 launay triangulation of the DFN. The parallelized subsurface flow and reactive transport code
158 PFLOTRAN [48] is used numerically integrate the governing flow equations to steady state. An
159 extension of the WALKABOUT particle tracking method [44, 45] is used to determine pathlines
160 through the DFN and simulate solute transport. Details of the suite, its abilities, applications, and
161 references for detailed implementation are provided in [27].

162 ## III.   GRAPH REPRESENTATIONS

163   We construct a graph representation of each DFN based on the network topology. Let $F = \{f_i\}$
164 for $i = 1, \ldots, n$ denote a DFN composed of n fractures. We define a mapping, ϕ, that transforms F
165 into a graph $G(V, E)$ composed of $n = |V|$ vertices, and $m = |E|$ edges. For every $f_i \in F$, there is
166 a unique vertex $u_i \in V$,

$$\phi : f_i \to u_i. \tag{16}$$

167 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 corre-
168 sponding vertices,

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

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

$$\phi^{-1} : G' \to F', \tag{18}$$

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

175   We also include source s and target t vertices into G to incorporate flow direction. Every
176 fracture that intersects the inlet plane $\mathbf{x}_0$ is connected to the source vertex,

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

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

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

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

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

187   Figure 2 shows the graph obtained from the fracture network shown in Fig. 1 using the mapping
188 φ. The source vertex is colored red and the target vertex is colored blue. The geometric layout of
189 the graph is an arbitrary projection into $\mathbb{R}^2$ using a force-directed layout algorithm [49]. Though
190 the positions do not represent actual fracture locations in $\mathbb{R}^3$ the drawing gives some perspective
191 on the connectivity of the fracture network and the graph path lengths between the source and
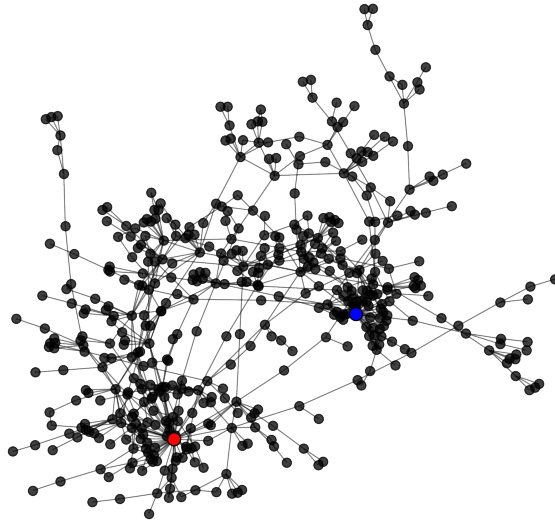192 target.

Figure 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.

A. 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 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 number of edges between the source and the target. The pre-image of this subgraph, which is its equivalent fracture subnetwork F', has the fewest number of 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 of all edges $|U|/|E|$ in the graph as a function of k. Thin semi-transparent lines are individual network realizations and 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,
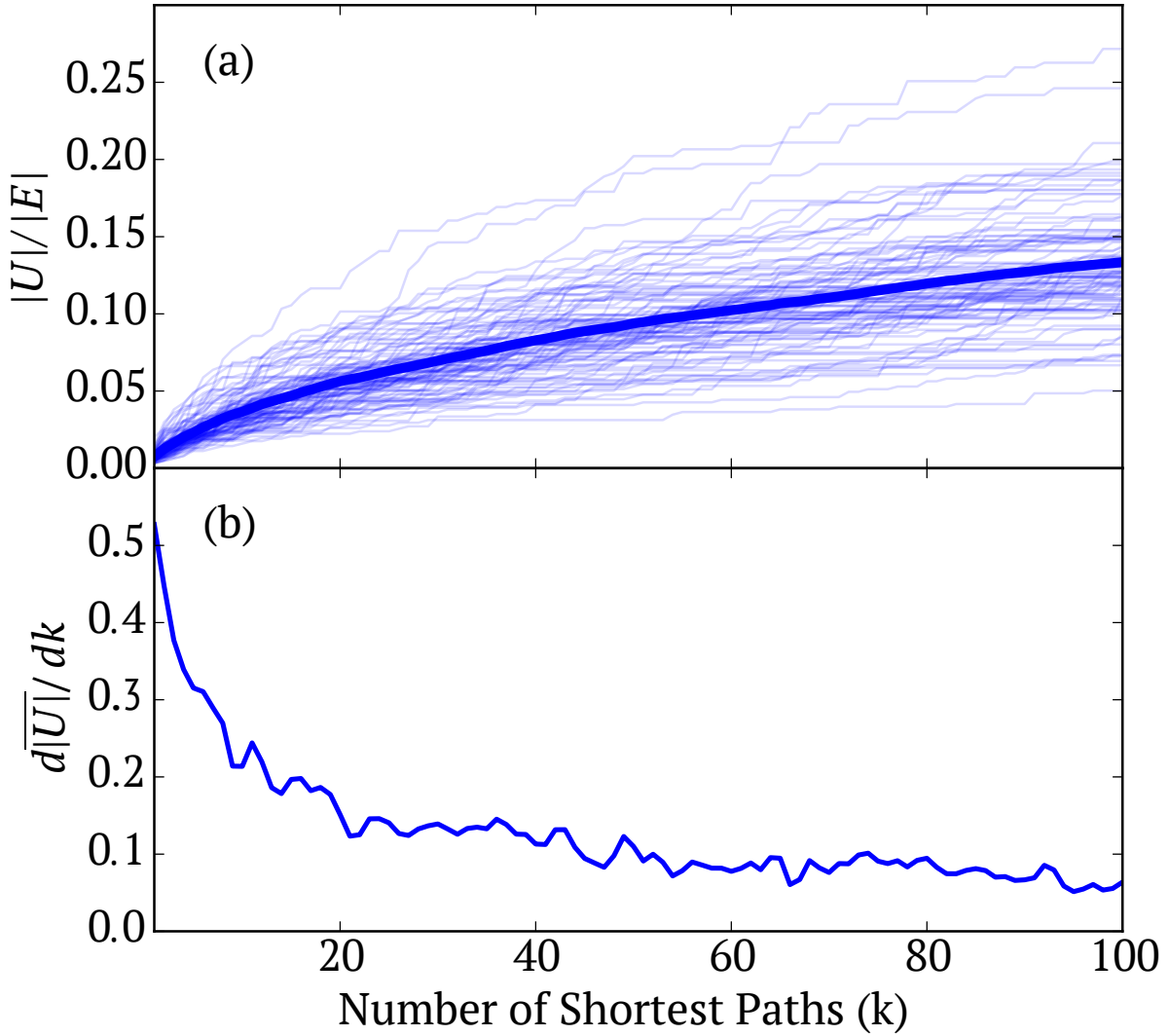
Figure 3. (a) The fraction of edges $|U|/|E|$ in the graph as a function of the number of shortest paths k. Thin semi-transparent curves are individual network realizations and the thick curve is the average of all networks. (b) The rate of change of the for the average size $\overline{|U|} = \sum|U|/100$ 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.

225 which is the case in two-dimensions. If the line of intersection between two fractures aligns with
226 the pressure gradient there will be a gradient within the dead-end fractures and thus flow. Hence,
227 the presence of dead-end fractures changes the local flow field on intersecting fractures and thus
228 its removal does as well. The 2-core typically makes up between 50%-60% of the graph edges
229 (not shown in Fig. 3). The source and target vertices are always retained in the 2-core.

230   For a graph *G* with *n* vertices and *m* edges the shortest-path set can be computed using breadth-

231 first search in $O(m+n)$ time [51]. The computation of the *k* shortest paths is harder but still can

232 be done in polynomial time, $O(kn(m+n\log n))$ [52]. Computing the *k*-core composition has time

233 complexity of $O(m)$ [53]. The subgraphs sets are computed using the NETWORKX graph software

234 package [54].

235   Figure 4 shows three subnetworks (top) and their subgraphs (bottom) derived from the network

236 and graph shown in Fig. 1 and Fig. 2. Semi-transparent vertices indicate fractures that have been

237 eliminated from the fracture network. The full network is made of 481 fractures, the shortest

238 path is made of 3 fractures (left), the ten shortest paths contain 23 fractures (middle), and the 2-

239 core contains 276 fractures (right). This reduction in number of fractures drastically changes the

240 number of cells in the mesh used for flow and transport simulations. The full network is meshed

241 with 910397 triangles, the shortest path is meshed with 5438 triangles, the ten shortest paths are

242 meshed with 69353 triangles, and the 2-core is meshed with 639319 triangles.

243   The method to obtain first passage times using these subnetworks can be conceptually divided

244 into the four steps: (i) A graph representation of a DFN is constructed using the mapping defined in

245 (16) and (17); $\phi : F \rightarrow G$. (ii) A subgraph composed of the *k* shortest paths between the source and

246 target is identified; $G' \subseteq G$. (iii) We isolate the subnetwork that is the pre-image of the extracted

247 subgraph; $\phi^{-1} : G' \rightarrow F'$. (iv) To obtain estimates of first passage times through each subnetwork,

248 they are subject to the same meshing procedure, initial and boundary conditions for the pressure,

249 and particle tracking initial conditions as the full network; equations (7) through (14).

IV.   METHOD PERFORMANCE

251   We measure the method's performance in terms of accuracy and efficiency. First, we compare

252 predictions of the first passage time in the full network and those obtained using each subnetwork.

253 Second, we compare the computational cost for the simulations.

A.   First Passage Times

255   Accuracy of the method is determined by comparing the first passage times (14) in F and F'

256 and statistics of the distribution of first passage times (15) for the ensemble of networks $\Omega = \{F\}$

257 and their subnetworks $\Omega' = \{F'\}$. Let $\mathcal{L}$ denote the operator that takes a DFN F as an input and
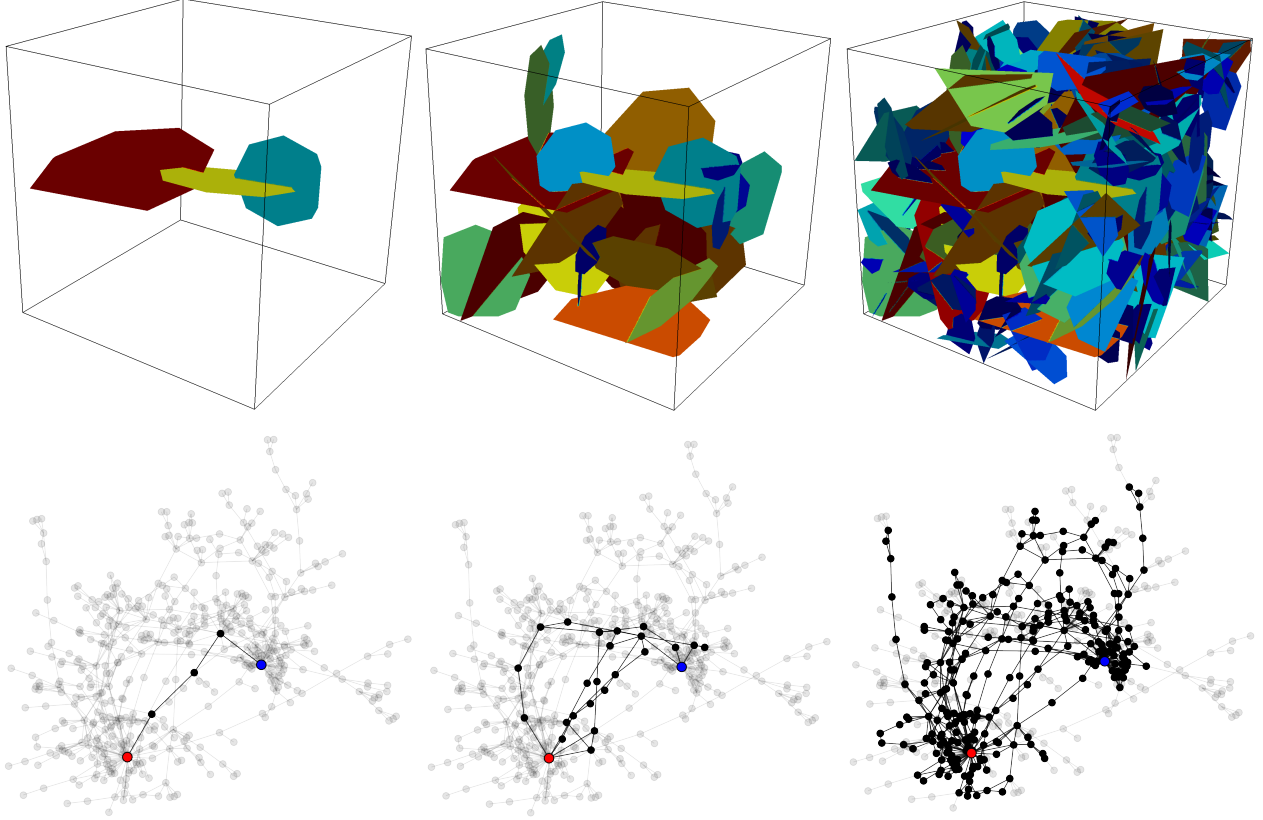
13

Figure 4. Subnetworks (top) and subgraphs (bottom) derived from the network and graph shown in Fig. 1 and Fig. 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. Semi-transparent vertices denote fractures that have been eliminated from the fracture network.

258 returns the first passage time $\hat{\tau}$,

$$\mathcal{L} : F \rightarrow \hat{\tau}. \tag{21}$$

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

$$\mathcal{L} : F' \rightarrow \hat{\tau}'. \tag{22}$$

260 The goal is that the error

$$\|\hat{\tau} - \hat{\tau}'\| \tag{23}$$

261 is small for each realization and in terms of their distributions obtained from Ω and Ω'. Individual
262 realizations provide a single value of $\hat{\tau}'$ to directly compare with $\hat{\tau}$. The distributions of $\hat{\tau}'$ and $\hat{\tau}$ are
263 compared in terms of their first two moments. Differences between the distributions are measured
264 by computing the Kullback-Leibler divergence (relative entropy), smaller values of which indicate

14

265 better agreement between the two probability densities. The two-sample Kolmogorov-Smirnov

266 test is also used to determine whether these differences are statically significant.

267   Figures 5(a)-(e) show $\hat{\tau}'$ obtained in the subnetworks plotted against $\hat{\tau}$. The (a) shortest paths,

268 (b) five shortest paths, and (c) ten shortest paths are on the top row and the (d) twenty shortest

269 paths, (e) 2-core are on the bottom. Values are divided by the median passage time of the en-

270 semble of particles through all one hundred networks to non-dimensionalize time. The black line

271 corresponds to identical first passage times in the subnetworks and the full networks. Deviations

272 are quantified by computing the coefficient of determination R^2. Values of R^2 that are closer to one

273 indicate better agreement between $\hat{\tau}'$ and $\hat{\tau}$ over the set of sample networks (values are provided

274 in Table I). In general, the first passage times of the shortest network (blue) are close to actual

275 first passage times, but there are exceptions ($R^2 = 0.54$). Values that deviate from the trend occur

276 in networks where the particle with the earliest passage time does not travel along the shortest

277 topological path. There is less scatter in the comparison with the five shortest paths ($R^2 = 0.57$),

278 but there are still outliers. Including the ten shortest paths leads to much better agreement between

279 with the full network ($R^2 = 0.86$). However, the increase in accuracy by increasing to include the

280 twenty shortest paths is less than going from five to ten ($R^2 = 0.90$). The difference between the

281 first passage times through the 2-core and the full networks is very small ($R^2 = 0.99$).

282   Figure 5 (f) shows the distributions of $\hat{\tau}$ and $\hat{\tau}'$ (15) obtained from the subnetworks and the full

283 network. Table I reports the first two moments of the distributions (mean: μ; variance: σ^2) of $\hat{\tau}$ and

284 $\hat{\tau}'$ along with the results of the two-sample Kolmogorov-Smirnov test and Kullback-Leibler diver-

285 gence measure. In general, there is good agreement between the distributions obtained using the

286 subnetworks and the full network. The distribution of $\hat{\tau}'$ in the shortest path networks has a higher

287 mean, a longer tail and higher variance than the full network. As more paths are included into

288 the subnetworks the Kullback-Leibler (KL) divergence measure decreases indicating better agree-

289 ment between with the distribution of first passage times in the full network. The 2-core matches

290 the full network values well for all values. The two-sample Kolmogorov-Smirnov test rejects the

291 null hypothesis that the distributions $\hat{\tau}'$ from the shortest paths are from the same distribution of

292 $\hat{\tau}$ obtained for the full network. The first two moments of the distributions of $\hat{\tau}'$ through the five,

293 ten, and twenty shortest paths and 2-core are close to those of $\hat{\tau}$. The two-sample Kolmogorov-

294 Smirnov test returns low values of the KS statistic and high p values for the five, ten, and twenty

295 shortest paths. Comparison of the full network with the 2-core resulted in lower KS values and

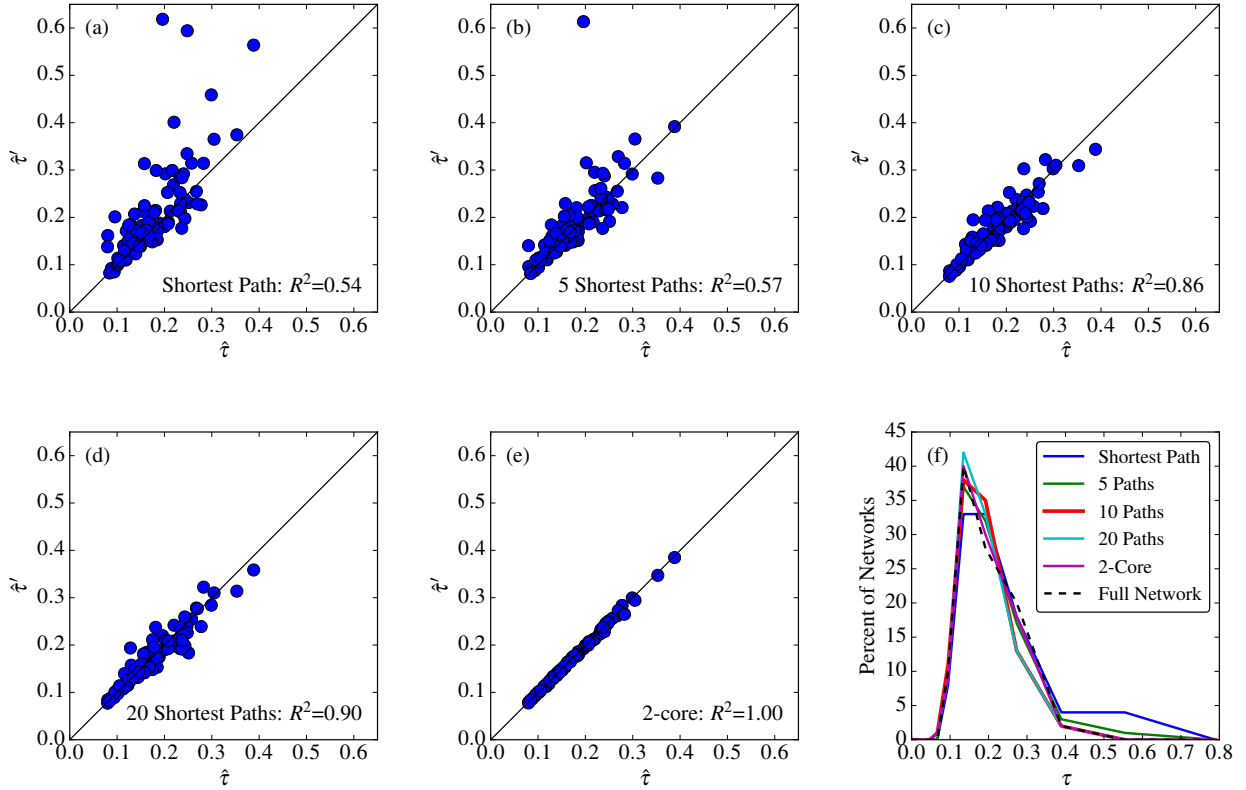296 higher p values than any other subnetwork.

15

Figure 5. First passage times in subnetworks vs the first passage time through the full network. Time has been non-dimensionalized by the median breakthrough of the ensemble of particle through all one hundred 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 one 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, (e) Two-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 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 outlet plane. The five shortest paths contain a few more

16

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

Subnetwork	μ	σ^2	KS	p-value	KL
Shortest Path	0.202	9.73×10^{-3}	0.140	0.261	6.35×10^{-2}
5 Shortest Paths	0.185	6.02×10^{-3}	0.080	0.894	3.01×10^{-2}
10 Shortest Paths	0.173	3.19×10^{-3}	0.080	0.894	1.59×10^{-2}
20 Shortest Paths	0.171	3.21×10^{-3}	0.080	0.894	6.99×10^{-3}
2-Core	0.173	3.51×10^{-3}	0.050	0.999	1.03×10^{-3}
Full Network	0.175	3.61×10^{-3}			

305 fractures and show an increase in surface area. On average, the ten shortest paths contain dou-
306 ble the number of fractures as the shortest path but less than double the surface area. Because
307 the shortest path is contained within the ten shortest paths, these values further indicate that the
308 shortest paths are composed of larger fractures. The 2-core subnetworks contain $\approx 56\%$ of the
309 total number of fractures and $\approx 75\%$ of the surface area. Thus, the compliment of the 2-core, the
310 dead-end fractures, make up half of the network by number and consist of mostly small fractures.

Table II. Mean values for geometric observables in the subnetworks identified using subgraphs. No. F: Number of fractures, SA: Network surface area [m^2], No. Cells: Number of computational cells. ($\cdot$) are the percentages of these values when compared to the mean values of the total network.

No.

Subnetwork	No. F	(%)	SA [m^2]	(%)	No. Cells	(%)
Shortest Path	8.55	(1.91)	6.12×10^2	(6.54)	1.99×10^4	(2.44)
5 Shortest Paths	12.05	(2.69)	8.12×10^2	(8.66)	2.90×10^4	(3.56)
10 Shortest Paths	17.55	(3.92)	1.08×10^3	(11.54)	4.43×10^4	(5.44)
20 Shortest Paths	25.69	(5.74)	1.43×10^3	(15.25)	6.51×10^4	(8.07)
2-Core	247.78	(55.39)	7.07×10^3	(75.40)	5.44×10^5	(66.88)
Full Network	447.34	(100.00)	9.37×10^3	(100.00)	8.13×10^5	(100.00)

311   The consequences of these reduced mesh sizes with respect to computational time are provided
312 in Table III and shown in Fig. 6. The average required wall clock time for the three primary
313 sections of the DFNWORKS workflow (network meshing, pressure simulation, and transport simu-
314 lation) are provided. Computations are performed using a server that has 64 cores; 1.4 GHz AMD
315 Opteron(TM) Processor 6272 with 2048 KB of cache each. Meshing is performed in parallel using
316 either 16 cores or the total number of fractures, whichever is less. The flow solution is determined
317 using 16 cores. Transport is performed using a single core. The time required for network gener-
318 ation prior to meshing is not included in the comparison because it is required for every network
319 and subnetwork, but is on the order of one second per network. The computation of the shortest
320 paths, *k* shortest paths, and 2-core subgraphs using NETWORKX take less than one second each
321 and those times are also omitted.

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)

322   In terms of total run time, the *k* shortest paths subnetworks are over an order of magnitude
323 faster than the full network. The networks contain significantly fewer fractures to mesh, which
324 results in fewer degrees of freedom in the linear system of pressure and faster solver convergence.
325 The time required for transport is also drastically reduced because a smaller number of fractures
326 intersect the inlet plane and thus fewer particles are inserted into the domain. The time required
327 for the 2-core is less than that for the full network, but is the same order of magnitude.
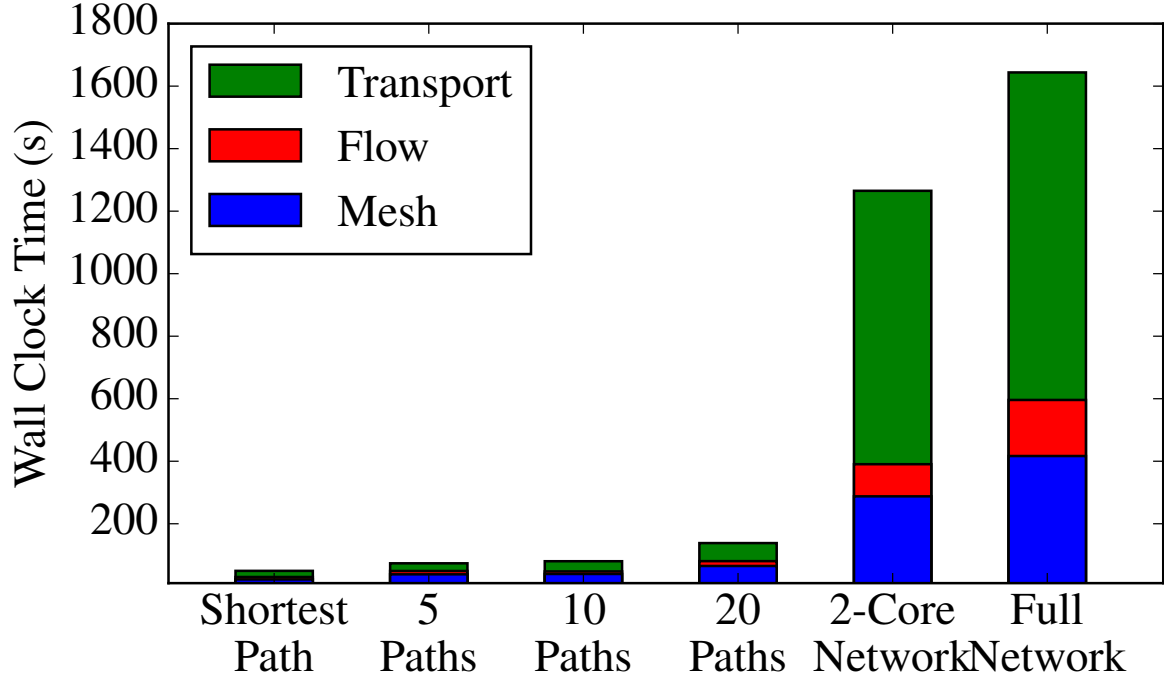
18

Figure 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.

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 pre-image 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 number of edges between the source and the target, correspond to the fewest number of 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

19

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 Figures 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 Figure 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 twenty 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 ten and twenty 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 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 filed 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 semi-generic 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 fractures 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 macro-scale structure of the network and the direction of flow. In dense networks, meso-scale features, e.g. fracture permeability, and micro-scale 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.

ACKNOWLEDGEMENTS

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

[2] C. Jenkins, A. Chadwick, and S. D. Hovorka, Int. J. Greenh. Gas. Con. **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. Energ. **147**, 500 (2015).

[6] National Research Council, *Rock fractures and fluid flow: contemporary understanding and applications*, edited by U. C. on Fracture Characterization and F. Flow (National Academy Press, 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, *Final report of the migration in a single fracture: experimental results and evaluation* (Nat. Genossenschaft fd Lagerung Radioaktiver Abfälle, 1985).

[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.-Sol. Ea. **117** (2012).

[13] A. Frampton and V. Cvetkovic, Water Resour. Res. **47** (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** (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** (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, Frontiers in Physics **1**, Art (2013).

[32] S. M. Hope, P. Davy, J. Maillot, R. Le Goc, and A. Hansen, Frontiers in Physics **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 T. Vis. Comput. Gr. (2016).

[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. Bogdanov, V. Mourzenko, J.-F. Thovert, and P. Adler, Phys. Rev. E **76**, 036309 (2007).

[38] J.-R. de Dreuzy, P. Davy, and O. Bour, Water Resour. Res. **38** (2002).

[39] A. Frampton and V. Cvetkovic, Water Resour. Res. **46** (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** (2009).

[42] J. Boussinesq, J. Math. Pures Appl **13**, 21 (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, Computat. Geosci. **19**, 1123 (2015).

[45] S. L. Painter, C. W. Gable, and S. Kelkar, Computat. 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, (LaGriT) Los Alamos National Laboratory," http://lagrit.lanl.gov (2013).

[48] 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*, Tech. Rep. ((Report No.: LA-UR-15-20403) Los Alamos National Laboratory, 2015).

[49] T. M. J. Fruchterman and E. M. Reingold, Software: Practice and Experience **21**, 1129 (1991).

[50] S. B. Seidman, Soc. Networks **5**, 269 (1983).

[51] R. Sedgewick and K. Wayne, *Algorithms* (Addison-Wesley, 2016).

[52] J. Yen, Management Science **17**, 712 (1971).

[53] V. Batagelj and M. Zaversnik, CoRR **cs.DS/0310049** (2003).

[54] A. A. Hagberg, D. A. Schult, and P. Swart, in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, Vol. 2008 (2008) pp. 11–16.