Computers & Geosciences 50 (2013) 59–71

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

# Synthetic benchmark for modeling flow in 3D fractured media

Jean-Raynald de Dreuzy [a,b,*], Géraldine Pichot [c], Baptiste Poirriez [d], Jocelyne Erhel [c]

[a] CNRS, Géosciences Rennes, Campus de Beaulieu, 35042 Rennes cedex, France
[b] Institute of Environmental Analysis and Water Studies, CSIC, Barcelona, Spain
[c] INRIA, Campus de Beaulieu, 35042 Rennes cedex, France
[d] IRISA, Campus de Beaulieu, 35042 Rennes cedex, France

ARTICLE INFO

*Article history:*
Received 13 April 2012
Received in revised form
23 July 2012
Accepted 25 July 2012
Available online 3 August 2012

*Keywords:*
Fractured media
Single-phase flow
Benchmark
Stochastic model

ABSTRACT

Intensity and localization of flows in fractured media have promoted the development of a large range of different modeling approaches including Discrete Fracture Networks, pipe networks and equivalent continuous media. While benchmarked usually within site studies, we propose an alternative numerical benchmark based on highly-resolved Discrete Fracture Networks (DFNs) and on a stochastic approach. Test cases are built on fractures of different lengths, orientations, aspect ratios and hydraulic apertures, issuing the broad ranges of topological structures and hydraulic properties classically observed. We present 18 DFN cases, with 10 random simulations by case. These 180 DFN structures are provided and fully documented. They display a representative variety of the configurations that challenge the numerical methods at the different stages of discretization, mesh generation and system solving. Using a previously assessed mixed hybrid finite element method (Erhel et al., 2009a), we systematically provide reference flow and head solutions. Because CPU and memory requirements stem mainly from system solving, we study direct and iterative sparse linear solvers. We show that the most cpu-time efficient method is a direct multifrontal method for small systems, while conjugate gradient preconditioned by algebraic multrigrid is more relevant at larger sizes. Available results can be used further as references for building up alternative numerical and physical models in both directions of improving accuracy and efficiency.



## 1. Introduction

Intensity and localization of flows in fractured media have been intensively discussed in the various perspectives of high-level nuclear waste storage, oil and gas production including lately the non-conventional shale gas and water resources in hard-rock aquifers (National Research Council, 1996; Neuman, 2005). Because flows are essentially concentrated in the fractures, the flow pattern is embedded in the fracture network structure and even in a subset of it when accounting for fracture-scale channeling effects (de Dreuzy et al., 2001c; Tsang and Neretnieks, 1998). Inheriting both from the topological complexity of the network and from the fracture aperture heterogeneity, flows are widely distributed and highly correlated. Their complexity has promoted a wide variety of different modeling frameworks including equivalent continuous media (Long et al., 1982; Tsang et al., 1996), dual porosity media (Warren et al., 1963) and Discrete Fracture Networks (DFNs) (Cacas et al., 1990; Dverstop and Andersson, 1989).

A DFN represents a fractured medium as a network of interconnected elements generated according to the statistical fracture properties issued by field observations (mostly coming from outcrops, wells and tunnels) (Billaux et al., 1989; Cacas et al., 1990; Dershowitz and Einstein, 1988; Dverstop and Andersson, 1989; Hsieh, 1998; Xu and Dowd, 2009). The DFN is made up of an ensemble of statistical laws describing the main fracture characteristics as size, orientation, transmissivity, density (Bourbiaux, 2010; Davy et al., 2006b; Davy et al., 2010; National Research Council, 1996). DFNs are built as fundamental analogs of natural media by sharing the same discrete structure and the same statistical properties as those observed on sites. DFNs are subsequently used as the basis for simulating mechanical, hydraulic or transport processes both for direct and inverse problems (Davy et al., 2006a; de Marsily et al., 2005; Doughty et al., 1994; Le Goc et al., 2010; Sudicky and McLaren, 1992; Verscheure et al., 2012). They have been applied for the transport of radionuclides from High-Level Radioactive Wastes (HLRW), the stability of vitrified nuclear wastes, the mechanical stability of rock blocks or the production of oil from fractured reservoirs (Baghbanan and Jing, 2007; Blocher et al., 2010; Bourbiaux, 2010; Cacas et al., 1990; Crevoisier et al., 2011; Long et al., 1982; Wellman et al., 2009).

* Corresponding author at: CNRS, Géosciences Rennes, Campus de Beaulieu, 35042 Rennes cedex, France.
*E-mail address:* jr.dreuzy@gmail.com (J.-R. de Dreuzy).

0098-3004/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2012.07.025

The interest of DFNs comes from their high similarity with the key features of natural fractures including the wide variety of their size and hydraulic properties (Bonnet et al., 2001; Bour and Davy, 1999). The simultaneous occurrence of fractures on a wide range of scales and their arrangement in intricate clusters challenge the classical simulation methods of physical and chemical processes (de Dreuzy and Erhel, 2002; Frampton and Cvetkovic, 2007; Kalbacher et al., 2007). The power-law distribution of fracture lengths is likely the most constraining feature for modeling as it rules out the a priori existence of a characteristic homogenization scale and requires any model to cover a wide range of scales (Bonnet et al., 2001). The stochastic nature of modeling also enhances the variety of encountered connectivity configurations. While DFNs are inherently 3D structures, most flow and transport studies have been conducted in 2D. Indeed, 3D simulations face numerous modeling issues including the generation of good quality meshes for the complex fracture network structures, the treatment of the large-scale numerical systems issued by discretization and the broad range of fracture lengths and transmissivities (Bonnet et al., 2001; Davy et al., 2010; Schmittbuhl et al., 1993). The need of robust and highly efficient numerical methods is even reinforced by the large sets of random simulations necessary to get reasonable estimates of uncertainties. While several numerical methods have been developed, they are still to be compared underlining the need of a common benchmark (Cacas et al., 1990; Dershowitz and Fidelibus, 1999; Gylling et al., 1998; Lenti and Fidelibus, 2003; Long et al., 1985; Maryka et al., 2004; Nordqvist et al., 1992; Vohralik et al., 2007).

We present in this article a comprehensive benchmark for flow simulations in DFNs. The benchmark proposes reference head and flow solutions in a wide variety of fracture networks and is designed to serve several purposes. It is built up on a series of structures organized by increasing level of complexity for setting up numerical methods. Elementary configurations are amenable to analytical or well-known numerical solutions and can be used as toy models in the early phase of development. Advanced configurations are made geologically relevant to assess numerical models, like pipe networks or mappings on 3D grids. In the pipe network approach, fracture flow is approximated by a mesh of 1D pipes at the network scale, either displayed randomly or relating fracture intersections (Cacas et al., 1990; Dershowitz and Fidelibus, 1999; Gylling et al., 1998; Nordqvist et al., 1992). These simplified models do not involve numerical difficulties, but their relevance cannot be a priori ascertained, especially when mixing fractures of different sizes. In the mapping approach, fractures are projected on a regular 3D mesh (Fourno et al., 2004; Tsang et al., 1996). If the major advantage is the ultimate possibility to account for matrix flow, fine grids are necessary to represent accurately fracture intersections and flows. Thus, even for few fractures, this model issues a very large numerical system. Thanks to recent progress in the simulation of 3D DFNs (Erhel et al., 2009a; Pichot et al., 2012; Pichot et al., 2010), we propose reference solutions obtained with a fine model and an accurate numerical finite element method.

We successively describe the domain configuration, flow model and boundary conditions (Section 2), the DFN test cases (Section 3), the numerical methods used to establish reference head and flow fields for the advanced cases (Section 4), the characteristics of the networks and the simulation results (Section 5). Section 6 compares the performances of generally available system solvers, as linear system solving is the most CPU-intensive step of the simulation chain. All test cases presented hereafter are available as a supplementary material to this article. File organization, contents and format are described in Appendices A and B.

## 2. Domain definition, flow equation and boundary conditions

In order to define a bounded open domain, the network is embedded into a cube of edge size $L$, orientated along the directions of a $x,y,z$ coordinate system with the origin at the center of the cube. The matrix is considered impervious, thus the flow domain is the union of the $N_F$ fractures $\Omega_f$ ($f=1\ldots N_F$), with $N_I$ intersection $S_k$ ($k=1\ldots N_I$) between the fractures. In each fracture $\Omega_f$, the diffusion equation is approached by:

$$\begin{aligned}&\mathbf{u}=-T\nabla h\\&\nabla\times\mathbf{u}=0\end{aligned}\tag{1}$$

where $\mathbf{u}$ is the velocity multiplied by the aperture of the fracture, $h$ is the head, $T$ is an equivalent transmissivity, generally given by the local aperture through Poiseuille's law (Auradou et al., 2005; Brown, 1987; Méheust and Schmittbuhl, 2001). The model must be completed with continuity conditions on fracture intersections $S_k$, which are written:

$$h_{k,f}=h_k,\forall f\in F_k\sum_{f\in F_k}\mathbf{u}_{k,f}\times\mathbf{n}_{k,f}=0\tag{2}$$

with $F_k$ the set of fractures intersecting on $S_k$, $h_k$ the head on the intersection $S_k$, $h_{k,f}$ the trace of the head on $S_k$ in the fracture $f$, $\mathbf{u}_{k,f}$. $\mathbf{n}_{k,f}$ the normal velocity through the intersection $S_k$ coming from the fracture $\Omega_f$ (Erhel et al., 2009a; Noetinger and Jarrige, 2012; Vohralik et al., 2007). Boundary conditions are classical permeameter boundary conditions: two opposite faces of the cube have fixed charges (Dirichlet type boundary conditions) and the four orthogonal faces are impervious (Neumann type boundary conditions). Moreover, there is no flow between fractures and matrix. Boundary conditions on the fracture $f$ are summarized:

$$\begin{aligned}&h=h_+\ \text{on}\ \Gamma_f\cap\Gamma_{y+}\\&h=h_-\ \text{on}\ \Gamma_f\cap\Gamma_{y-}\\&\mathbf{u}\times\mathbf{n}=0\ \text{on}\ \Gamma_f\backslash(\Gamma_{y+}\cup\Gamma_{y-})\end{aligned}\tag{3}$$

where $\Gamma_{x-},\Gamma_{x+},\Gamma_{y-},\Gamma_{y+},\Gamma_{z-},\Gamma_{z+}$ are the six faces of the cube and $\Gamma_f$ is the border of the fracture $f$. The direction of the head gradient along $y$ will be referred as the main flow direction.

## 3. DFN test cases

The benchmark is made up of fracture networks of increasing complexity divided in four categories of elementary fracture networks for which analytical solutions are available, 2D-like fracture networks that are direct extensions of 2D DFNs, 3D percolation-like structures and fully 3D power-law DFNs.

### 3.1. Analytical solutions of flow in elementary fracture networks

We call elementary "analytical" fracture networks those simple networks on which flows can be calculated analytically. They remain simple and are basically extensions to 3D of the linear head solution on a 1D medium. The first example is made up of a single fracture of homogeneous transmissivity orientated along one of the directions ($x$ or $z$) orthogonal to the main flow direction ($y$), fully connected to the $y_-$ and $y_+$ Dirichlet boundary conditions and passing through the center of the cube (Fig. 1a). Within the fracture, head depends linearly on the distance to the boundary condition. For the case of a fracture parallel to the $z$ direction, the linear evolution of head from $h_-$ to $h_+$ from $y_-$ to $y_+$ leads within the fracture plane to:

$$h(x,y,z)=h_-+\sqrt{\frac{\left(\frac{xy}{L}+\frac{x}{2}\right)^2+\left(\frac{y^2}{L}+\frac{y}{2}\right)^2}{x^2+y^2}}(h_+-h_-)\tag{4}$$

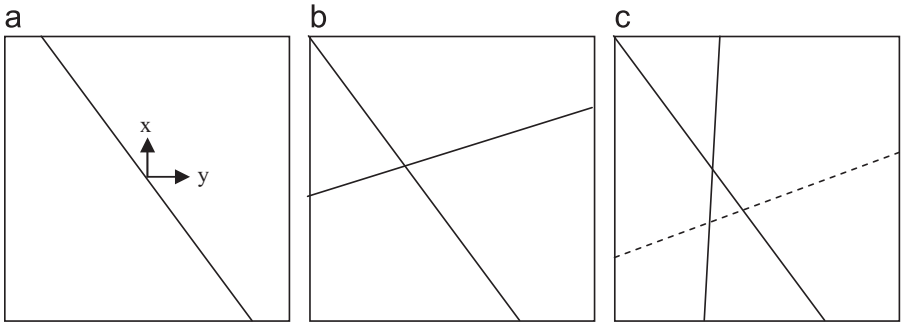


**Fig. 1.** Fracture network configurations for which head field can be determined analytically. All lines are traces of 3D fractures parallel to the $z$ direction.

where it is implicitly assumed that the point of coordinates $(x,y,z)$ belongs to a plan passing through (0,0,0).

This case can be straightforwardly extended to two fracture cases of interest for validating assessing numerical methods. The first one consists in adding a fracture orientated along the same direction orthogonal to the $y$ direction as the first fracture and connected to the no-flow boundary conditions (Fig. 1b). In this additional fracture, flow is uniformly null. The second one consists in adding the same kind of fracture orientated along the same direction orthogonal to the $y$ **direction as the first fracture passing through the cube center and connected to the fixed-head boundary conditions (Fig. 1c). In each fracture, the flow remains linearly dependent on the distance to the boundary condition within the fracture plane as given by (4).

Several test networks can be built up on this configuration for example on the two fractures case of Fig. 1b by adding other fractures touching the no-flow boundary conditions parallel to the same direction as the other ones (dashed line on Fig. 1b). Another possible extension is that of the regular grid. It should be underlined that the grid should remain aligned with the coordinate system. If the grid is tilted, heads can no longer be computed analytically because of the introduction of a variety of shortcuts that breaks the former independence of flow solution in the different fractures. The only exception is the 45° rotation for which an analytical solution can be built up thanks to the invariance of the zigzag pattern by translation orthogonally to the main flow direction along the $x$ direction. The head field depends only on the distance to the $y_+$ boundary condition along the fracture network:

$$h(x,y,z) = h_- + \left(\frac{y}{L} + \frac{1}{2}\right)(h_+ - h_-) \tag{5}$$

### 3.2. 2D-like fracture networks

2D-like fracture networks are direct generalizations of the previous test networks. They display a translational invariance in the $z$ direction, which makes them similar to a 2D network with the traces of fracture in the $(x,y)$ plane. Flows computed with the 3D network can thus be compared with the flows computed with the 2D network. We provide two test cases in the benchmark, made up of, respectively 4 and 50 fractures (Fig. 2). Moreover, ten random networks are generated for each test case, yielding a benchmark of 20 DFNs. Heads on intersection points of the 2D networks are given in single text files D2_LIKE_h-no, the content of which is provided by Table 4, and results for the 3D corresponding networks are discussed in Section 4. Even if the 2D reference remains numerical, it is a very accurate solution without discretization error as the mesh is the network itself and head solution between two nodes is linear (de Dreuzy and Erhel, 2003; de Dreuzy et al., 2001b). The only error, which arises when solving the linear system, can be easily controlled. We have used the direct multifrontal method implemented in SuiteSparse (Davis and Duff, 1999) to solve the linear system.

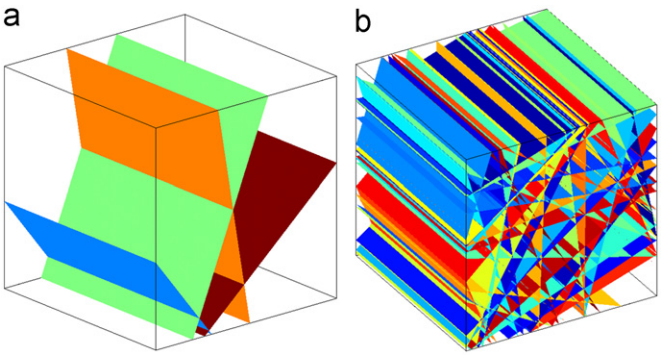


**Fig. 2.** 2D-like fracture networks with (a) 10 and (b) 50 fractures.

### 3.3. Percolation-like structures

Random networks of small fractures having all the same size $l_{\min}$ display a well-defined collective behavior that has been revealed by percolation theory (Stauffer and Aharony, 1992). If the percolation threshold is sensitive to the fracture shape (de Dreuzy et al., 2000), the scaling of the geometrical and hydraulic properties at threshold only depend on the Euclidean dimension. It especially means that the scaling of the equivalent permeability $K_{eq}$ at threshold follows a universal law:

$$K_{eq}(L) \sim L^{-\mu/\nu} \tag{6}$$

where $\mu/\nu$ is a very well defined exponent that can however only be approached numerically. Approximations on site percolation structures of $\mu/\nu$ are $2.26 \pm 0.04$ for Normand and Herrmann (1995) and 2.283 for Kozlov and Lagues (2010). The test consists in computing the equivalent permeability at different scales ranging typically over two orders of magnitude and further evaluating $\mu/\nu$. It may be used for assessing simplification methods based for example on equivalent pipe network concepts (Cacas et al., 1990). The exponent is sensitive to systematic errors in the local approximations necessary to transform local connectivity structures into equivalent permeability structures and can be used as a first necessary step to qualify approximations.

### 3.4. 3D power-law DFNs

We consider a series of DFNs defined by the following ensemble of characteristics (Table 3). Fracture centers are uniformly distributed within the domain. Fractures are ellipses of

aspect ratio $e$ (major axis over minor axis) uniformly distributed between 0.25 and 0.75 to avoid disk-like and needle-like fractures. Fracture lengths $l$ follow a power-law distribution $f_l$ such as:

$$f_l(l) \sim l^{-a_{3D}} \tag{7}$$

with a characteristic exponent $a_{3D}$ equal to 2.7 or 3.7 (Bonnet et al., 2001). The minimum fracture major axis is fixed to $l_{min}$. Fracture lengths are thus distributed from $l_{min}$ to $\alpha L$, where we recall that $L$ is the characteristic domain scale, and $\alpha$ is a coefficient larger than 1, accounting for the truncation of ellipses larger than the cube. Fracture orientations are either uniformly distributed or close to three orthogonal directions, with allowed deviations to the orthogonal directions of at most 10°. Fracture local apertures $a$ are either constant equal to $10^{-4}$ m/s or have the generally observed Gaussian distribution, with self-affine correlation function $f_a$ (Méheust and Schmittbuhl, 2001; Méheust and Schmittbuhl, 2003):

$$f_a(a) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(a-\mu)^2}{2\sigma^2}} & \text{if } a \geq a_c \\ 0 & \text{if } a \leq a_c \end{cases} \tag{8}$$

where $a_c$ is a small aperture value designed to prevent the occurrence of negative or null aperture values (Méheust and Schmittbuhl, 2003). In practice $a_c/\mu = 10^{-3}$ is enough. The mean aperture $\mu$ is kept at $10^{-4}$ m/s and the standard deviation $\sigma$ is equal to the mean leading to a so-called closure $\sigma/\mu$ of 1. Local fracture transmissivities $T$ are related to the fracture aperture $a$ by assuming locally the validity of Poiseuille's law (Taylor, 1953):

$$T = \frac{g}{v}\frac{a^3}{12} \tag{9}$$

with $v$ the dynamic viscosity and $g$ the classical gravitational acceleration. Taking for the fluid water at ambient temperature, $T = 8.2\ 10^{-7}$ m$^2$/s for $a = 10^{-4}$ m/s. Finally, fracture density is counted according to the appropriate definition of the percolation parameter $p$ for power-law distributions of ellipses (de Dreuzy et al., 2000):

$$p = \frac{N}{L^2} \int (\min(l, \alpha L))^3 \ f_l(l) \ dl \tag{10}$$

where $N$ is the number of fractures. We normalize it by its value at threshold $p_c$ and consider two characteristic values of $p/p_c$ just above percolation threshold ($p/p_c = 1.2$) and twice above percolation threshold ($p/p_c = 3$). The full set of DFNs is obtained from the 16 combinations of the two choices of length, orientation and aperture distributions and of the two possible values of density. Each network is named with a combination of the labels given in Table 3. For example, the case of a network with uniform orientations (ORUNI), a power-law distribution of fracture lengths with $a_{3D} = 2.7$ (A27), just above threshold (THRE), with self-affine correlated fracture apertures (TSAFF) is named ORUNI_A27_THRE_TSAFF.

Because of the statistic characterization of fracture properties, modeling is inherently stochastic. We provide in the benchmark 10 random samples generated for each set of parameters. On top of conceptual relevance, random simulations also broaden the range of network configurations on which numerical methods are assessed. The 160 fracture networks are made available with the H2OLab platform, under the format documented in Appendix A1.

The length distribution has a major influence on the network structure as shown by the differences between the left and middle columns of Fig. 3. When the power-law length exponent $a_{3D}$ increases from 2.7 to 3.7, the range of fracture lengths contributing to connectivity widens as well as the tortuosity and the density of dead ends. Structures close to percolation threshold promote high degrees of network scale channeling and solute dispersion (de Dreuzy et al., 2001a; Le Goc et al., 2009; Renard and Allard, 2012; Tsang and Neretnieks, 1998). At higher densities (right column of Fig. 3), even though connectivity becomes less critical, hydraulic and transport properties can still strongly differ from homogeneous like structures. Indeed, a limited number of main flow paths is in interaction with a dense network of smaller fractures. Fracture aperture distributions enhance this multiple porosity like hydraulic behavior (de Dreuzy et al., 2001c; de Dreuzy et al., 2002). Preferential fracture orientations have, respectively less influence (Fig. 3, top line compared to bottom line). They however control to

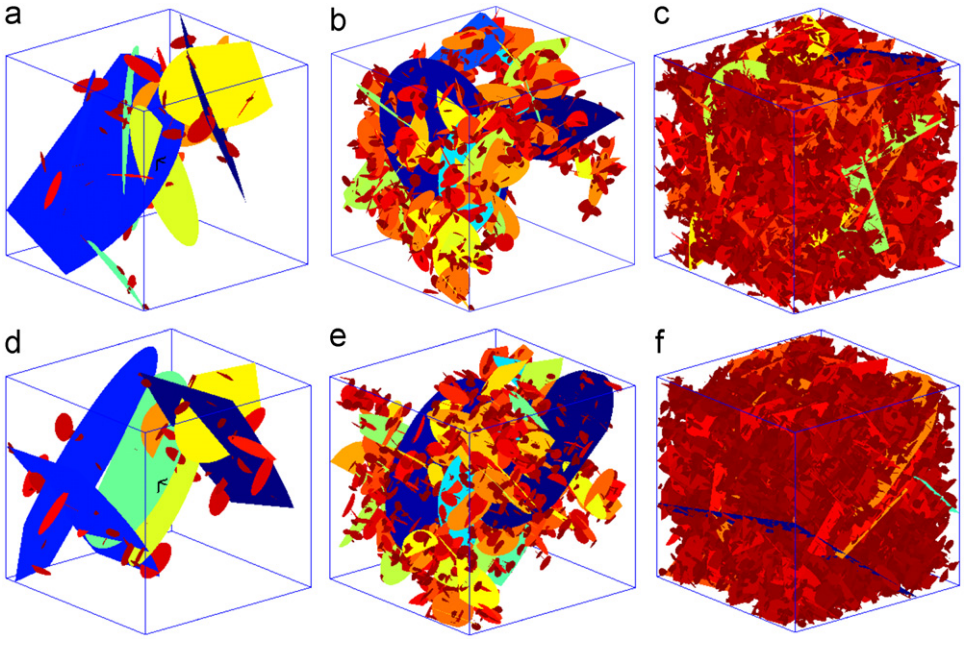


**Fig. 3.** Examples of tested networks. Only clusters connected to both $y_-$ and $y_+$ (inflow and outflow faces) are considered. Fracture length increases from red to blue. a) ORUNI_A27_THRE, (b) ORUNI_A37_THRE, (c) ORUNI_A37_DENS, (d) ORTHO_A27_THRE, (e) ORTHO_A37_THRE and (f) ORTHO_A27_THRE. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

some extent the disposition of fracture intersections within the fracture planes that constrain the generation of high-quality meshes (Erhel et al., 2009a; Kalbacher et al., 2007). Note that the difference between uniformly orientated networks and orthogonal fracture orientations is only the fracture direction, fracture centre, whereas major and minor axes remain identical for each fracture (Fig. 3, top line compared to bottom line). If some of the largest fractures and the general connectivity structures are similar, connectivity details still vary substantially.

## 4. Numerical methods for complex DFNs

For the DFNs described above, no simplified model can be a priori devised and numerical methods are required to approximate the head and flow fields. We have developed a complete software suite, called MP_FRAC, which generates a random DFN and simulates a steady-state flow in this network, with various boundary conditions (Erhel et al., 2009a). This software is integrated in the platform H2OLab (Erhel et al., 2009b). In order to generate random DFNs, with the various probability laws modelling the geometry and the physical properties, a specific tool is developed in $H_2$OLAB (Erhel et al., 2008; Erhel et al., 2009b), with streams of random numbers generated by the RngStream package (L'Ecuyer et al., 2002), and input and output parameters described in XML files. This generic tool allows also running multi-parametric simulations with a large number of samples. It provides simulation results for each sample as well as statistical results. The software MP_FRAC uses intensively this tool, whereas computational geometry is handled by the CGAL package (CGAL, Computational Geometry Algorithms Library, http://www.cgal.org). Currently, MP_FRAC can simulate steady-state flow with various boundary conditions. We first describe the numerical model implemented and in the next section, we provide our simulation results for the networks of Section 3.4. It should be noted that our model can also be used for all the networks of Section 3.

### 4.1. Numerical model

The numerical model is based on the Mixed Finite Element method, mainly for two reasons: it ensures both local and global mass conservation and it provides an accurate velocity field, which can be used in subsequent transport simulations. We implemented the so-called RT0 scheme (Brezzi and Fortin, 1991; Raviart and Thomas, 1977). The networks considered have a very specific geometry: it is a 3D intricate structure of 2D domains. Since the matrix is impervious, the mesh is 2D inside each fracture, 1D at the intersections between fractures, and a 3D set of 2D intersecting domains at the network scale. A first difficulty is to generate this mesh, since it cannot be handled directly by a mesh generator. A second difficulty is to ensure head and flow continuity at the intersections of the fractures and a third challenge is to solve the resulting linear system.

To generate the mesh, a first approach is to discretize first the boundaries and the intersections, then the 2D fractures. However, this method induces very small angles because of the intricate geometry and may fail for some networks (Mustapha, 2005). Therefore, we designed a new method, introducing a pre-processing step where fracture borders and intersections are discretized using a regular 3D grid. These staircase-like discretizations are finally projected onto each fracture plane. Then a good quality 2D mesh is generated in each fracture plane, using the resulting 1D discretizations of the border and intersections. Local adjustments are necessary to guarantee geometrical properties. We developed this approach in both a conforming and a non conforming settings (Erhel et al., 2009a; Pichot et al., 2012; Pichot et al., 2010). Local modifications and a non conforming method are also used in (Vohralik et al., 2007).

With a hybrid method and a conforming mesh, it is finally quite easy to ensure the continuity conditions at the intersections, because of the choice of the main unknowns (the trace of head on each edge of the mesh) (Erhel et al., 2009a). With a non conforming mesh, we used the Mortar framework to write the discrete problem (Pichot et al., 2010). However, the pre-processing step induces particular cases where some parts of intersections are common to three fractures or more. Thus, we had to generalize the Mortar method to deal with these configurations (Pichot et al., 2012).

Either in the conforming or the non conforming case, linear equations written at each edge of the mesh express local mass conservation. The resulting linear system $Ax=b$, with $x$ the trace of head on edges and $b$ accounting for boundary conditions, is large. It has as many unknowns as edges in the mesh but is sparse, with roughly five non zero coefficients per line for a mesh with triangles (Erhel et al., 2009a). The matrix $A$ of the system is SPD (symmetric positive definite), also for the non conforming case. Thus several solving algorithms can be used: a direct method, based on the Cholesky factorization; an algebraic multigrid method; a preconditioned conjugate gradient method, with various preconditioners; a domain decomposition method (Poirriez, 2011). High performance computing is required to handle very large systems. Once the system is solved, it is easy to compute the mean head, as well as the transverse flux through each edge and the velocity inside each triangle, using the RT0 scheme.

### 4.2. Assessment of the model

The Mixed Finite Element method is well studied in the literature (Roberts and Thomas, 1991). Here, we introduce additional approximations with the pre-processing step and with the generalized Mortar method. Although we did not prove theoretical results, we checked convergence experimentally and observed convergence behaviour with an order 1, both in the conforming and the non conforming cases (Erhel et al., 2009a; Pichot et al., 2012; Pichot et al., 2010).

In order to validate the software, we generated the simple networks described in the first two categories. We got a very good agreement with analytical results. Regarding 2D-like networks, each intersection in the 3D network corresponds to a crossing point $i$ in the 2D network and the head value at any edge $j$ of the intersection should be equal to the head value at the crossing point. In order to compare 2D and 3D results, we compute the relative matrix of errors $e_{ij}$. For the 20 tested 2D-like fracture networks, this matrix is very small, as expected.

Regarding general complex networks, we compute the equivalent permeability $K_{eq}$, defined by the rate of the specific flow ($Q/L^2$) to the head gradient (($\Delta h/L$):

$$K_{eq} = \frac{Q_{\text{in}}}{L\Delta h} \tag{11}$$

with $Q_{\text{in}}$ the flow integrated over all edges belonging to the inflow boundary. Normal fluxes on each edge and mean head on each triangle are computed from the trace of head, using the Raviart-Thomas basis functions (Brezzi and Fortin, 1991). Equivalent permeability is in any case a relevant first-order measure for upscaling hydraulic properties in complex structures as it is sensitive to both the topological structures and the hydraulic heterogeneities that has extensively used for qualifying flows in fractured media (de Dreuzy et al., 2010; Long et al., 1982).

In our numerical simulations, we check systematically both local and global mass conservations. The norm of the residual $r=b-Ax$ measures the deviation from an exact local conservation rule. It is small with an accurate linear solver. The deviation from

**Table 1**
List of 2D like DFNs with their name (corresponding to the folder name where results are stored), number of fractures, fracture length, transmissivity and orientation distribution.

| Name | Number of fractures | Length $l$ | Transmissivity ($T$) | Orientation |
|---|---|---|---|---|
| D2_like_N10<br>D2_like_N50 | 10<br>50 | $l \gg L$ | $T=8.2\ 10^{-7}$ m$^2$/s | Uniform |

a global conservation rule is measured by computing all the fluxes at the borders of the cubic domain. More precisely, no flow boundary conditions are checked and a global criterion at the domain scale is defined by:

$$\mathrm{prec} = \frac{Q_{\mathrm{in}} + Q_{\mathrm{out}}}{Q_{\mathrm{in}}} \tag{12}$$

with $Q_{\mathrm{out}}$ the flow integrated over the outflow boundary. All our simulation results exhibit a very small residual and a very small global error. Thus our software MP_FRAC is very reliable and can be used for studying statistical properties of DFNs.

Finally, we would like to emphasize that our method is very robust, in the sense that we could generate many random 3D networks, generate the mesh of these DFNs and simulate the flow, with no failure at all. With the software MP_FRAC, we could simulate flow in more than $10^5$ complex 3D fracture networks.

We provide now comprehensive results for the networks presented in Section 4, which can be used as a reference for comparison with other models Table 1.

## 5. Complex 3D DFN cases with reference head and flow fields

We use the numerical methods of Section 4 to compute reference head and flow fields on the 2D-like networks of Section 3.2 and on the power-law DFN networks cases of Section 3.4. As already stated, 10 random simulations have been run for both 2D-like networks and each of the 16 power-law DFNs. Physical and numerical parameters are given in Table 2. Computations are done with 64-bit floating point arithmetic. For all the networks, the mesh generated is conforming and is based on a mesh step equal to 0.1 m. The characteristics of the networks are given in Table 2. Results are obtained on a bi-cpu six-core Intel Xeon 3 GHz computer with 96 GB of RAM, configured with windows server 2008 64-bit system. The reference head and flow fields are computed using either a direct solver or an iterative PCG solver with a convergence criterion of $10^{-14}$ (Table 2). More precisely, two solver libraries are interfaced: SuiteSparse, a direct solver (Davis and Duff, 1999) and PCG, of Hypre suite, with default parameters, a conjugate gradient preconditioned by one cycle of boomer-AMG (Falgout et al., 2005). The choice of the linear solver is discussed in Section 6.

### 5.1. Main characteristics of networks and some results

Some integrated results are given in Table 5, as means over the 10 samples generated. The set of 3D power-law DFNs cover a wide range of configurations, at least partly characterized by the number of fractures (n_frac), the cumulated fracture surface (surface), the mean intersection length (int_length) and the average distance between two intersections (int_dist) (Table 5).

The mean number of fractures varies from 180.2 to 6845. The cumulated fracture surface differs by at most 25% for the two power-law exponents, showing that the length distribution has a stronger impact on the network structure than the cumulated surface. Mean intersection length depends mostly on the length distribution (int_length $\approx$ 0.6 for $a_{3D}=3.7$ and int_length $\approx$ 1.3 for $a_{2D}=2.7$). The mean number of edges (edges), which is also the system size, is quite large, from 3.66 million to 27.8 million. It depends highly on the density and to a lesser extent on the power-law exponent Tables 3 and 4.

The orientation has only a small influence on the equivalent permeability $K_{eq}$, which is surprisingly only weakly sensitive to the aperture distribution. It is true as long as the correlation length of the fracture aperture distribution is not much larger than the distance between fracture intersections (de Dreuzy and Méheust, 2012).

Local mass conservation is ensured by a small residual in the system solver and the criterion *prec*, testing global mass conservation, is very small. Time measurements and performance results are discussed in Section 6.

**Table 2**
Geometrical, hydraulic and numerical parameters common to all tested networks. Units are indicative. Dimensionless parameters can straightforwardly be deduced by taking the scale $l_{\mathrm{min}}$ as the reference scale.

| | Parameter name | Description | Value |
|---|---|---|---|
| Geometrical and hydraulic | $L$ | System size | 20 m |
| | $l_{\mathrm{min}}$ | Minimum fracture major axis | 1 m |
| | $h_+$ | Maximum head (imposed on $\Gamma_f \cap \Gamma_{y+}$) | 20 m |
| | $h_-$ | Minimum head (imposed on $\Gamma_f \cap \Gamma_{y-}$) | 0 m |
| | | Head gradient | 1 |
| Numerical | | Maximum mesh length | 0.1 m |
| | | System solver | Direct or PCG |
| | | Solver precision in PCG | $10^{-14}$ |

**Table 3**
DFN characteristics and names.

| Characteristics | Options | Label |
|---|---|---|
| Position | Uniform distribution in domain | |
| Shape | Uniform distribution of major to minor axes in [0.25,0.75] | |
| Orientation | Uniform distribution | ORUNI |
| | Nearly orthogonal fracture orientations | ORTHO |
| Length | Power law distribution with $a_{3D}=2.7$ | A27 |
| | Power law distribution with $a_{3D}=3.7$ | A37 |
| Density | 20% above threshold ($p/p_c=1.2$) | THRE |
| | Twice above threshold ($p/p_c=3$) | DENS |
| Aperture | Uniform equal to $10^{-4}$ m/s ($T=8.2\ 10^{-7}$ m$^2$/s) | TUNIF |
| | Self affine correlated, Gaussian distribution ($\mu=10^{-4}$ m/s,$\sigma/\mu=1$) | TSAFF |

**Table 4**
D2_LIKE_h-no file format and content.

| **D2_LIKE_h-no** | |
|---|---|
| Column name | Description |
| *no* | Line number within the file (not necessary) |
| *no_inter* | 2D node number |
| *x* | *x* coordinate of the node |
| *y* | *y* coordinate of the node |
| *h* | Head value at the node |

### 5.2. Detailed description of networks

For each of these 180 fracture networks, three levels of intermediary results are provided at different stages of the modeling process covering, respectively (1) the network intersections and fracture contours, (2) a possible conforming mesh, (3) and the linear systems issued by a Mixed Hybrid Finite Element scheme. These

three levels can be used for addressing different modelling issues. The single fracture network description can be used for evaluating simple estimates like those given by effective media (Kirkpatrick, 1971) or percolation (de Dreuzy et al., 2001b; Stauffer and Aharony, 1992), the fracture and intersection characteristics for generating meshes, the mesh for finite element schemes and the linear systems for linear system solvers. In Appendix A, we document precisely these outputs.

### 5.3. Reference head and flow

Head and flux files formats are given in Appendix B. Visual inspection of the flow for two close networks (Fig. 4) show similar head fields but quite different flows. Head similarities come from the fixed boundary conditions. Depending on local head gradients, flows are more sensitive to the fracture aperture distribution as shown by the comparison of the top and bottom lines of Fig. 4. Fig. 4 further shows the multi-scale nature of channelling at the fracture and network scales.

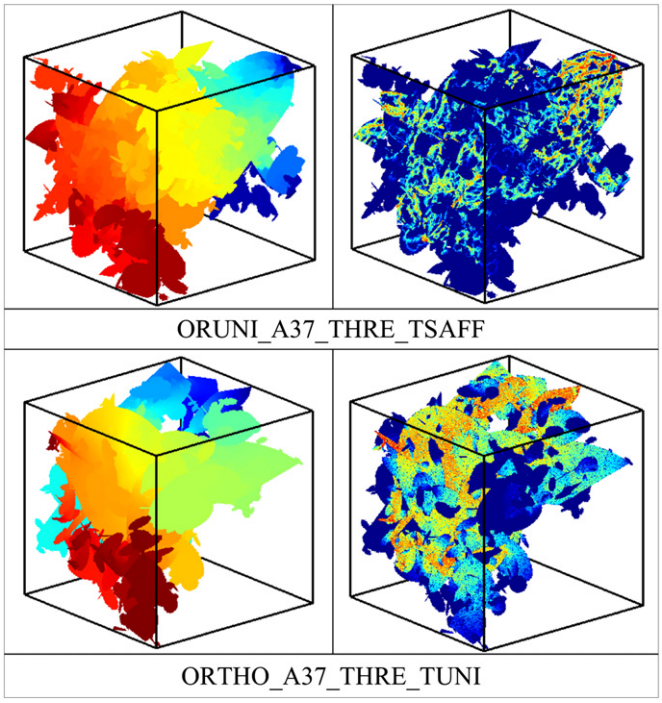


**Fig. 4.** Head (left column) and average flow (right column) for two of the test networks. Heads and flows increase from black to red on linear and logarithmic scales, respectively from their minimal to the maximal values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Aside from the detailed head and fluxes given in each mesh cell within the fractures, we also provide the jump of flux at the fracture intersections. This output may be used more straightforwardly to benchmark equivalent pipe network approaches (Cacas et al., 1990; Nordqvist et al., 1992) for very large domains, that are currently out of reach of the methods presented in Section 4. Jumps of flux at intersections may also be used as a starting point to build approximate solute transport solutions. The full fracture network may be approached as a graph with the nodes being the fracture intersections and the edges the main flow tubes within the fracture determined by simple geometrical rules. The graph is directed and weighted by the fluxes at the intersections distributed among the edges. Such methods may be interesting in hydro-geophysics methods where only a rough approximation is necessary for assessing connectivity structures (Dorn et al., 2011).

## 6. Performance analysis

As can be observed from the simulation results (Table 5), the most CPU-intensive computation is the linear system solver, which is the critical step of the simulation, especially for the large fracture networks. Indeed, all other steps require in general a small amount of CPU time, increasing in most cases linearly with the system size (Erhel et al., 2009a). In Table 5, discretization time is high with 4 networks, with an exponent $a_{3D}=3.7$ and a high density, because of many local adjustments. In order to reduce this time, we are improving the algorithms and we will use the Mortar method with cheaper local adjustments (Pichot et al., 2012; Pichot et al., 2010).

The CPU time for solving the linear system increases with the system size, so that it is essential to use a fast sparse solver. In this section, we show that we could consider complex DFNs resulting in linear systems with millions of unknowns (Poirriez, 2011). In order to study thoroughly the system solvers, we provide results with a preliminary experiment and test cases inducing relatively small systems (Poirriez, 2011).

**Table 5**
Network results. All results are averages over 10 random realizations. Computational time for solving the linear system is given for the two solvers UMFPACK (tsUMF) and PCG (tPCG). Results are obtained on a bi-cpu six-core Intel Xeon 3 GHz computer with 96 GB of RAM, configured with windows server 2008 system.

| Network name | n_frac | surface | int_length | int_dist | edges | edges_i | Keq | Prec | t_disc | t_mesh | t_sysprep | tsPCG | tsUMF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D2_LIKE_N4 | 4 | 1,635.73 | 20 | 18.8223 | 8.34E+06 | 1,062.2 | 6.8E−08 | 3.84E−13 | 0.03 | 3 | 4.,3 | 21.2 | 55 |
| D2_LIKE_N50 | 49,9 | 18,798.5 | 20 | 5.2 | 4.73E+07 | 150,092 | 8.60E−07 | 1.00E−13 | 22.9 | 50.1 | 83.2 | 1171.1 | 745.7 |
| ORTHO_A27_DENS_TSAFF | 808.3 | 5,186.4 | 1.4 | 2.9 | 1.31E+07 | 35,408.1 | 1.30E−07 | 1.10E−11 | 51.2 | 28.4 | 32.7 | 615.4 | 295.9 |
| ORTHO_A27_DENS_TUNIF | 808.3 | 5,186.4 | 1.4 | 2.9 | 1.31E+07 | 35,408.1 | 1.90E−07 | 3.70E−13 | 48.4 | 27.4 | 34.3 | 545.6 | 386.5 |
| ORTHO_A27_THRE_TSAFF | 136.3 | 1,499.7 | 1.2 | 5.4 | 3.66E+06 | 3,981.8 | 1.90E−08 | 7.80E−12 | 3.8 | 8.9 | 10.6 | 132.6 | 25.3 |
| ORTHO_A27_THRE_TUNIF | 136.3 | 1,499.7 | 1.2 | 5.4 | 3.66E+06 | 3,981.8 | 3.60E−08 | 4.10E−13 | 3.6 | 9.2 | 10.4 | 126.7 | 25.4 |
| ORTHO_A37_DENS_TSAFF | 5515.6 | 9,008 | 0.6 | 1.1 | 2.28E+07 | 98,018.4 | 8.10E−08 | 1.30E−11 | 695.7 | 33.7 | 44.1 | 907.8 | 979.8 |
| ORTHO_A37_DENS_TUNIF | 5426.6 | 8,851.2 | 0.6 | 1.1 | 2.24E+07 | 95,674.1 | 1.30E−07 | 6.90E−13 | 526.8 | 26.5 | 29 | 632.5 | 631 |
| ORTHO_A37_THRE_TSAFF | 521.6 | 1,725.4 | 0.6 | 1.6 | 3.66E+06 | 7,902.6 | 7.10E−09 | 4.80E−11 | 13.5 | 7.7 | 9.5 | 133 | 22.4 |
| ORTHO_A37_THRE_TUNIF | 521.6 | 1,725.4 | 0.6 | 1.6 | 3.66E+06 | 7,902.6 | 1.40E−08 | 2.10E−12 | 13.5 | 7.7 | 9.3 | 109.6 | 23.5 |
| ORUNI_A27_DENS_TSAFF | 965.5 | 6,058.5 | 1.4 | 2.3 | 1.54E+07 | 47,074.7 | 1.30E−07 | 6.30E−12 | 40.9 | 22.4 | 26.1 | 548.3 | 361.3 |
| ORUNI_A27_DENS_TUNIF | 965.5 | 6,058.5 | 1.4 | 2.3 | 1.54E+07 | 47,074.7 | 1.90E−07 | 2.00E−13 | 60.2 | 28.5 | 39.5 | 713.2 | 454.2 |
| ORUNI_A27_THRE_TSAFF | 180.2 | 1,876 | 1.3 | 3.9 | 4.69E+06 | 5,670.3 | 1.90E−08 | 2.00E−11 | 5.4 | 11.1 | 13.2 | 178.1 | 36.9 |
| ORUNI_A27_THRE_TUNIF | 180.2 | 1,876 | 1.3 | 3.9 | 4.69E+06 | 5,670.3 | 3.50E−08 | 1.10E−12 | 5.7 | 11.3 | 13.1 | 172.4 | 35.1 |
| ORUNI_A37_DENS_TSAFF | 6845 | 10,780.6 | 0.5 | 0.9 | 2.78E+07 | 131,359 | 8.60E−08 | 5.40E−12 | 930.2 | 40.8 | 56.1 | 1445.8 | 1154.8 |
| ORUNI_A37_DENS_TUNIF | 6845 | 10,780.6 | 0.5 | 0.9 | 2.78E+07 | 131,359 | 1.60E−07 | 5.80E−13 | 860.1 | 43.3 | 53.5 | 1417.6 | 1173.2 |
| ORUNI_A37_THRE_TSAFF | 727 | 2,157 | 0.6 | 1.4 | 4.74E+06 | 11,736.5 | 3.90E−09 | 1.70E−10 | 17.3 | 8.9 | 11.6 | 150.8 | 45.3 |
| ORUNI_A37_THRE_TUNIF | 727 | 2,157 | 0.6 | 1.4 | 4.74E+06 | 11,736.5 | 1.00E−08 | 5.50E−12 | 16.6 | 9.1 | 12.3 | 138.5 | 28.7 |

### 6.1. Preliminary experiment with small DFNs

In this preliminary experiment, we generate a large number of various random networks and solve the linear system with three different software libraries: SuiteSparse, PCG and Boomer-AMG, of Hypre suite, an algebraic multigrid solver, with default parameters (Falgout et al., 2005). We run sequential simulations on a machine with 4 quadri-core Intel Xeon 2.4 GHz processors and with 16 GB of RAM, configured with windows server 2003 64-bit system.

We define two sets or random DFNs, with a homogeneous transmissivity. In the first set, the system size increases with the ratio $N/N_c$, the number of fractures $N$ relatively to the number of fractures $N_c$ at percolation threshold. We choose three values of power-law exponents $a_{3D}$, a given mesh step, six values of the ratio $N/N_c$ and draw ten samples for each combination of parameters, yielding a set of 177 connected networks. Indeed, we do not consider non connected networks since the resulting linear system is singular. The system size ranges from 140,000 to 610,000. In the second set, the system size increases by refining the mesh. We fix the ratio $N/N_c$, choose three exponents $a_{3D}$ and nine mesh steps, and again draw ten samples for each combination, resulting in a set of 270 connected networks. The system size ranges from 60,000 to 5900,000. Finally, we get a total number of 417 connected networks, since some DFNs are in the two sets (Tables 6 and 7).

### 6.2. Results with the three sparse linear solvers

For the largest systems of this preliminary experiment, the direct solver fails because of memory requirements, so that 387 systems are solved among the 417 test cases. We could solve larger systems with CHOLMOD instead of UMFPACK (CHOLMOD will be interfaced very soon), but memory requirements would also increase rapidly with the size and would put a barrier. In the benchmark experiment (Table 5), the direct solver does not fail and is fast, thanks to the memory available (96 GB instead of 16 GB). As expected, we observe in both experiments an algorithmic complexity, with CPU time increasing as a power of the system size, but it is difficult to estimate the exponent because of the large variability and of the range considered.

Regarding AMG, an unexpected behaviour occurs, since convergence rate varies a lot from one system to another. When we simulate flow in 2D or 3D random heterogeneous porous media for example, the number of V-cycles of AMG is constant, resulting in a linear algorithmic complexity (Erhel et al., 2008; Erhel et al., 2009b). In Fig. 5, we plot the number of V-cycles for the first set of the preliminary experiment and observe slow convergence or even failure (when the number reaches the fixed limit of 500 cycles) for some networks, with a small residual only for 375 systems out of the 417 considered. We still investigate this result by characterizing the connectivity and the corresponding matrix graph. We could also tune the parameters in Hypre, but our simulations must be fast and reliable for any random network. Here, the matrix graph is particular, since the domain is neither a 2D surface nor a 3D volume but a 3D network of interconnected 2D subdomains. This particular structure could have an impact on the spectral properties of the iteration matrix, thus on convergence of the multigrid method. When the convergence rate is good, AMG is very fast, but it becomes expensive when convergence deteriorates. Thus we do not consider AMG in the benchmark experiment.

Finally, we use also PCG, preconditioned by one V-cycle of AMG, which turns out to be robust since all 417 systems can be solved accurately. In Fig. 6, we plot the number of PCG iterations for the first set of the preliminary experiment and observe a small number of iterations, with some variability, between 3 and 5 iterations. In the benchmark experiment, the number of iterations is also small, with some variability. In our simulations, convergence of PCG slows down slightly when the system size increases. Nevertheless, the combination of a Krylov method and a multigrid method is indeed very reliable and less memory consuming than a direct solver.

**Table 6**
Preliminary experiment: first set. The linear system size increases with the ratio $N/N_c$, the number of fractures $N$ relatively to the number of fractures $N_c$ at the percolation threshold.

| Parameter name | Description | Value |
|---|---|---|
| $L$ | System size | 3 m |
| $l_{min}$ | Minimum fracture major axis | 1 m |
| $a_{3D}$ | Power law exponent | 2.5; 3.5; 4.5 |
| $N/N_c$ | Relative number of fractures | 2; 3; 4; 5; 6; 7 |
| | Mesh step | 0.04 |
| | Number of random samples | 10 |

**Table 7**
Preliminary experiment: second set. The linear system size increases by refining the mesh.

| Parameter name | Description | Value |
|---|---|---|
| $L$ | System size | 3 m |
| $l_{min}$ | Minimum fracture major axis | 1 m |
| $a_{3D}$ | Power law exponent | 2.5; 3.5; 4.5 |
| $N/N_c$ | Relative number of fractures | 4 |
| | Mesh step | 0.01; 0.02; 0.03; 0.04; 0.05; 0.06; 0.07; 0.08; 0.09 |
| | Number of random samples | 10 |

### 6.3. Comparison of the three sparse linear solvers

In Figs. 7 and 8, we plot the CPU time of UMFPACK, AMG and PCG, versus the system size, for all DFNs in the preliminary

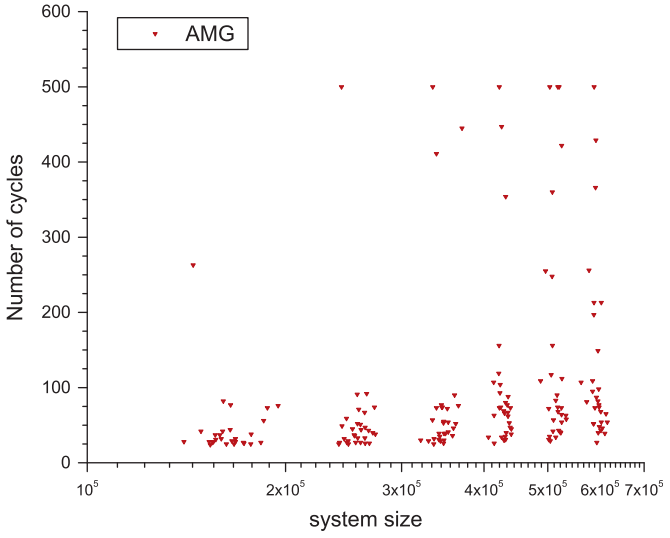


**Fig. 5.** Number of V-cycles in Boomer-AMG versus the system size for the first set of networks in the preliminary experiment, with a varying number of fractures.

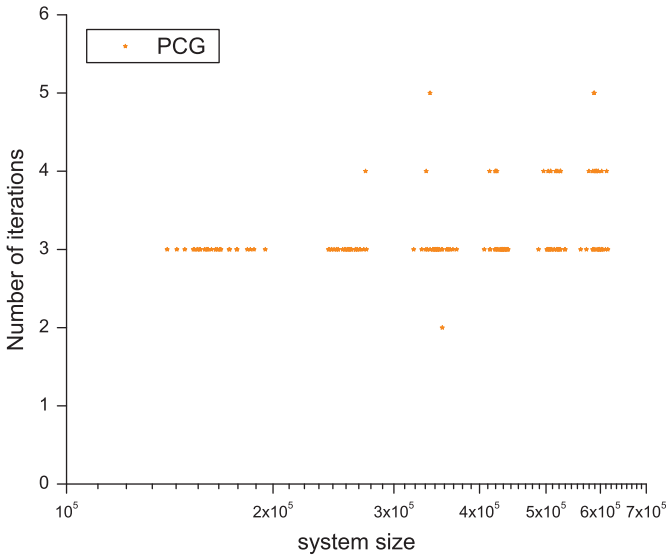


**Fig. 6.** Number of PCG iterations (CG preconditioned by Boomer-AMG) versus the system size for the first set of networks in the preliminary experiment, with a varying number of fractures.

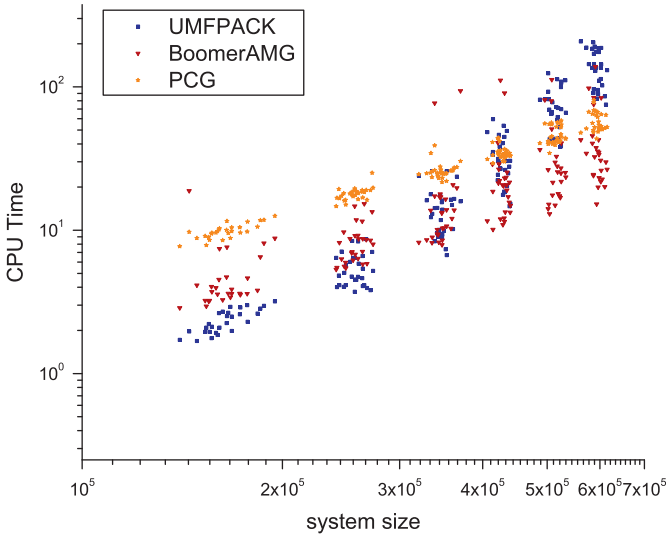


**Fig. 7.** CPU time of the three sparse solvers versus the system size, for the first set of the preliminary experiment, with a varying number of fractures.

experiments, with points missing when UMFPACK or AMG failed. The CPU time with UMFPACK and PCG for the benchmark experiment are given in Table 5.

The winner is AMG when it converges very fast but, as already pointed out in Section 6.2, it is not reliable in our preliminary experiment, since it converges very slowly for some DFNs. We have no clear explanation of this behavior, even though we reckon that the network structure may have an impact.

As can be expected, the direct solver is the most efficient up to a given size, which depends on the target computer. Clearly, memory requirements are a bottleneck for the direct solver. In the preliminary experiment with only 16 GB of RAM, the threshold is at about 0.5 million; with the benchmark experiment and 96 GB of RAM, the threshold is higher than 47 million.

In all our simulations, PCG (preconditioned by AMG) is reliable and requires less memory than a direct solver, thus both the direct and PCG solvers are our current methods of choice.

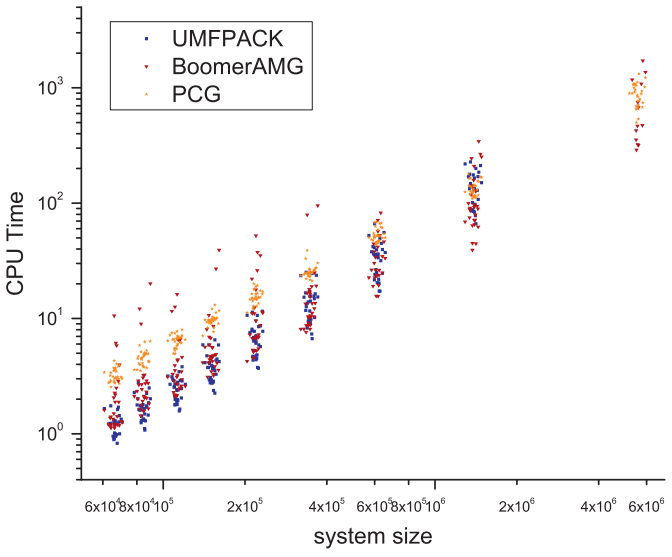


**Fig. 8.** CPU time of the three sparse solvers versus the system size, for the second set of the preliminary experiment, with a varying mesh step.

## 7. Conclusion and future work

We propose an extensive benchmark for modeling single-phase flow in stochastic Discrete Fracture Networks (DFNs). Analytical or highly accurate solutions are given for single fractures and 2D-like networks. Previously assessed mixed hybrid finite element methods (Erhel et al., 2009a) are used to propose reference solutions for geologically relevant DFNs. A set of 180 test cases is built on randomly generated DFNs, with fractures of different lengths, orientations, aspect ratios and hydraulic apertures issuing the broad ranges of topological structures and hydraulic properties classically observed. Structures and flow solutions are provided and fully documented. Both a direct and an iterative Preconditioned Conjugate Gradient methods are efficiently used to solve the large linear systems, of size up to 47 million. Available results can be used further as references for building up alternative physical and numerical methods in both directions of improving accuracy and efficiency.

The mesh generated in the benchmark is conforming but we plan to use also a non conforming mesh, thanks to our Mortar method (Pichot et al., 2012; Pichot et al., 2010). This should reduce the number of edges thus the memory and CPU requirements.

The model with a non conforming mesh will be combined with the domain decomposition method currently under development (Poirriez, 2011). This should also improve the performances, in particular with multicore or multiprocessor architectures.

Thanks to these numerical improvements, flow will be computed on very large DFNs, with many fractures and edges. The stochastic model will be used to run Monte Carlo simulations and to analyze statistical outputs. We aim at studying the equivalent permeability and at deriving upscaling rules.

## Acknowledgements

This project was funded by the Arc INRIA GEOFRAC. J.-R. de Dreuzy acknowledges the European Union for its additional funding through the IEF Marie-Curie fellowship (PIEF-GA-2009-251710).

## Appendix A. Database of DFN test cases

We document precisely the file list, structure and content for the elements given for the fracture networks. Directories are organized according to the network names of Table 5. For each simulation, the files listed below and summarized in Table 8 are stored in the same directory named "run_global_results" and the files Aij.txt and bij.txt are stored in a directory called "systems".

### *Fracture network*

Each fracture modelled by an ellipse is described by its centre, its normal, the unitary vector along the major and minor axes, the length of the major and minor axes (Table 9). Fracture centres are given in the reference coordinate system centred at the centre of the cube.

### *Fracture contours and intersections*

Fracture contours are ellipse polygons with segments described in file "polygon-no.vector" (Table 15). Intersections between fractures are segments given in a separate file "polygon-intersections-no.vector" (Table 16). The 2D coordinates within the corresponding fracture plane of those contour and intersections segments are given by the file "segments_coordinates_2D-no.vector" (Table 17).

### *Conforming mesh*

We provide a "nearly" conforming mesh issued by our numerical method (Erhel et al., 2009a). Indeed the projection step within each fracture plane, used to obtain a good quality mesh (see the meshing procedure presented in Section 4.2), introduces a small gap between the edges of the intersections in the intersected planes. But from the view point of the finite element method, the resulting mesh is conforming. We underline that this mesh is not unique but it can be used to test classical finite element methods. The mesh is described in three files for the triangles (Table 10), edges (Table 11) and vertices (Table 12). Coordinates of vertices are referenced within the fracture plane as the mesh has been generated within this 2D system. The coordinate system is centred at the centre of the ellipse with the three unitary axes along the major and minor axes of the ellipses ($\boldsymbol{u}$ $\boldsymbol{v}$) and the

**Table 8**
List of files.

| File name | Description | Column description |
|---|---|---|
| frac-no.vector | Fracture network | Table 9 |
| triangle-no.vector | Triangles of mesh | Table 10 |
| edge-no.vector | Edges of mesh | Table 11 |
| vertex-no.vector | Vertices of mesh | Table 12 |
| Aij.txt | Matrix of linear system | |
| bi.txt | rhs vector of linear system | |
| *h*.txt | head at the triangle center | |
| *q*.txt | flux on edges | |
| Jump_flux_by_frac_by_inter-no.vector | Jump of flux at intersections | Table 14 |
| Polygon-no.vector | Contour segments of the ellipse polygons | Table 15 |
| Polygon_intersections-no.vector | Intersection segments between ellipse polygons | Table 16 |
| Segments_coordinates_2D-no.vector | Contour segments of the ellipse polygons | Table 17 |

**Table 9**
Fracture network file (column number, parameter name and description).

| File frac-no | | |
|---|---|---|
| No | Name | Description |
| 1 | no | Line number within the file (not necessary) |
| 2 | Laxis_maj | Half length of fracture major axis |
| 3 | Laxis_min | Half length of fracture minor axis |
| 4 | c_x | Coordinates of fracture centre |
| 5 | c_y | |
| 6 | c_z | |
| 7 | n_x | Coordinates of vector normal to the fracture plane |
| 8 | n_y | |
| 9 | n_z | |
| 10 | no_fracture | Number identifying the fracture |
| 11 | u_x | Coordinates of vector along the fracture major axis |
| 12 | u_y | |
| 13 | u_z | |
| 14 | v_x | Coordinates of vector along the fracture minor axis |
| 15 | v_y | |
| 16 | v_z | |

**Table 10**
Characteristics of mesh triangles (column number, parameter name and description).

| File triangle-no | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | T | Transmissivity of the triangle |
| 3 | no_edge_0 | Number of the 3 edges defining the triangle |
| 4 | no_edge_1 | |
| 5 | no_edge_2 | |
| 6 | no_fracture | Fracture number to which belongs the triangle |
| 7 | no_triangle | Local triangle number in the fracture |
| 8 | no_vertex_0 | Number of the 3 vertices of the triangle |
| 9 | no_vertex_1 | |
| 10 | no_vertex_2 | |

**Table 11**
Characteristics of mesh edges (column number, parameter name and description).

| File edge-no | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | edge_no_global | Global edge number |
| 3 | edge_no_system | Position of the edge head unknown within the linear system |
| 4 | lim_arc | One if the edge belongs to the arc of an ellipse, 0 otherwise |
| 5 | lim_face | One if the edge belongs to a domain face, 0 otherwise |
| 6 | lim_intersection | One if the edge belongs to an intersection between fractures, 0 otherwise |
| 7 | lim_no_face | Border number to which belongs the edge if lim_arc is one, −1 otherwise |
| 8 | no_fracture | Fracture number to which belongs the point |
| 9 | no_triangle_0 | Triangles numbers to which belong the edge |
| 10 | no_triangle _1 | |
| 11 | no_vertex_0 | Vertices numbers defining the edge |
| 12 | no_vertex_1 | |

**Table 12**
Coordinates of mesh vertices (column number, parameter name and description).

| File vertex-no | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | coord_0 | Coordinates of the vertex in 2D within the fracture plane |
| 3 | coord_1 | |
| 4 | no_fracture | Fracture number to which belongs the point |
| 5 | no_vertex | Local vertex number in the fracture |

**Table 13**
Coding of cube faces.

| Number | Face |
|---|---|
| 0 | $x_-$ |
| 1 | $x_+$ |
| 2 | $y_-$ |
| 3 | $y_+$ |
| 4 | $z_-$ |
| 5 | $z_+$ |

normal to the ellipse plan ($\boldsymbol{n}$) forming a direct coordinate system. $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{n}$ are given for each fracture in the "frac-no" file (Table 9).

Additionally, each edge is given several labels identifying its possible location on ellipse arc, cube face and intersection (Table 11). All corresponding fields are equal to 0 for internal fracture edges. Edges can be simultaneously on a cube face and on an intersection or on an arc and on an intersection. Edges cannot however be simultaneously on an arc and on a cube face. Edges on a cube face are given the number of this cube face (Table 13). Edges on Dirichlet boundary conditions are assigned a fixed head, whether they are also intersections or not, while edges on ellipse arcs and other cube faces are assigned no flow boundary conditions. Intersection edges are listed in all the fractures to which they belong.

*Linear system*

The linear system $Ax=b$ is issued by the application of a mixed hybrid finite element scheme applied on the mesh described in the previous section and with the boundary conditions given in Section 2. The matrix $A$ is stored in the file Aij.txt, in coordinate format (line,column,value), with the number of non-zero elements written on the first line. The right-hand side $b$ is stored in the file bi.txt, with two columns $i$ and $val$ such that $b(i)=val$.

## Appendix B. Reference pressure and flow solutions

*Head and fluxes in mesh cells*

Mean head solution: file h.txt (fracture number, triangle number, head value). Mean head solution, constant in the corresponding triangle.

Flux solution: file $q$.txt (fracture number, edge number, triangle number, flux value). Normal fluxes through the three edges of each triangle.

*Jumps of flux at intersections*

Jumps of flux at each intersection of a given fracture are computed as an intermediary output between the full head or velocity field and the highly integrative equivalent permeability. They are stored in the file named Jump_flux_by_frac_by_inter-no.vector (Table 14). By convention, fluxes are positive when entering the fracture and negative otherwise. We underline that an intersection can be partly shared by more than two fractures after discretization. Thus, mass conservation is checked at the network level Table 15–17.

## Appendix C. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2012.07.025.

**Table 14**
Jump of flux at edges (column number, parameter name and description).

| File Jump_flux_by_frac_by_inter-no.vector | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | jumpflux | Jump of flux at the intersection |
| 3 | no_fracture | Fracture number of the intersection |
| 4 | no_inter | Intersection number |

**Table 15**
Contour segments of the ellipse polygons (column number, parameter name and description).

| polygon-no.vector | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | lim_arc | 1 if the segment belongs to the discretization of an ellipse arc, 0 otherwise |
| 3 | lim_face | 1 if the segment belongs to a domain face, 0 otherwise |
| 4 | no_face | Face number to which belongs the segment if lim_arc is one, −1 otherwise |
| 5 | no_fracture | Fracture number |
| 6 | no_segment | Global segment number |

**Table 16**
Intersection segments between ellipse polygons (column number, parameter name and description).

| polygon_intersections-no.vector | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | no_fracture1 | Fracture number to which belongs the intersection segment |
| 3 | no_fracture2 | Fracture number to which belongs the intersection segment |
| 4 | no_inter | Global number of the intersection |
| 5 | no_segment | Global segment number |

**Table 17**
Coordinates of intersections and contour segments of the ellipse polygons (column number, parameter name and description).

| segments_coordinates_2D-no.vector | | |
|---|---|---|
| 1 | no | Line number within the file (not necessary) |
| 2 | no_fracture | Fracture number to which belongs the segment |
| 3 | no_segment | Global segment number |
| 4 | point_0_coord_0 | Coordinates of the first extremity point P0 of the |
| 5 | point_0_coord_1 | segment in 2D within the fracture plane |
| 6 | point_1_coord_0 | Coordinates of the second extremity point P1 of the |
| 7 | point_1_coord_1 | segment in 2D within the fracture plane |

## References


Auradou, H., Drazer, G., Hulin, J.P., Koplik, J., 2005. Permeability anisotropy induced by the shear displacement of rough fracture walls. Water Resources Research 41, 9.

Baghbanan, A., Jing, L.R., 2007. Hydraulic properties of fractured rock masses with correlated fracture length and aperture. International Journal of Rock Mechanics and Mining Sciences 44 (5), 704–719.

Billaux, D., Chiles, J.P., Hestir, K., Long, J., 1989. 3-dimensional statistical modeling of a fractured rock mass—an example from the fanay-augeres mine. International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts 26 (3–4), 281–299.

Blocher, M.G., Cacace, M., Lewerenz, B., Zimmermann, G., 2010. Three dimensional modelling of fractured and faulted reservoirs: framework and implementation. Chemie Der Erde-Geochemistry 70, 145–153.

Bonnet, E., et al., 2001. Scaling of fracture systems in geological media. Reviews of Geophysics 39 (3), 347–383.

Bour, O., Davy, P., 1999. Clustering and size distributions of fault patterns: theory and measurements. Geophysical Research Letters 26 (13), 2001–2004.

Bourbiaux, B., 2010. Fractured reservoir simulation: a challenging and rewarding issue. Oil & Gas Science and Technology-Revue De L Institut Francais Du Petrole 65 (2), 227–238.
Brezzi, F., Fortin, M., 1991. Mixed and Hybrid Finite Element Methods. Springer, Berlin.
Brown, S., 1987. Fluid flow through rock joints: the effect of surface roughness. Journal of Geophysical Research 92 (B2), 1337–1347.
Cacas, M.C., et al., 1990. Modeling fracture flow with a stochastic discrete fracture network: calibration and validation. 1. The flow model. Water Resources Research 26, 3.
Crevoisier, D., Bouyer, F., Gin, S., 2011. Semi-stochastic generator (FRAGMA) of 2D fractured media by mechanistic analogy—Application to reactive transport in a fractured package of vitrified nuclear waste. Computational Materials Science 50 (4), 1387–1398.
Davis, T.A., Duff, I.S., 1999. A combined unifrontal multifrontal method for unsymmetric sparse matrices. ACM Transactions on Mathematical Software 25 (1), 1–20.
Davy, P., Bour, O., De Dreuzy, J.R., Darcel, C., 2006a. Flow in Multiscale Fractal Fracture Networks, 31–45.
Davy, P., Darcel, C., Bour, O., Munier, R., de Dreuzy, J.R., 2006b. A note on the angular correction applied to fracture intensity profiles along drill core. Journal of Geophysical Research-Solid Earth 111 (B11).
Davy, P., et al., 2010. A likely universal model of fracture scaling and its consequence for crustal hydromechanics. Journal of Geophysical Research 115 (B10411).
de Dreuzy, J.-R., de Boiry, P.P., Davy, P., G., 2010. Use of power-averaging for quantifying the influence of structure organization on permeability upscaling. Water Resources Research 46, W08519.
de Dreuzy, J.-R., Erhel, J., 2003. Efficient algorithms for the determination of the connected fracture network and the solution to the steady-state flow equation in fracture networks. Computers & Geosciences 29 (1), 107–111.
de Dreuzy, J.-R., and Y. Méheust. Influence of fracture scale heterogeneity on the flow properties of three-dimensional discrete fracture networks (DFN), Journal Geophysical Research-Earth Surface, submitted for publication.
de Dreuzy, J.R., Davy, P., Berkowitz, B., 2001a. Advective transport in the percolation backbone in two dimensions. Physical Review E, 64.
de Dreuzy, J.R., Davy, P., Bour, O., 2000. Percolation threshold of 3D random ellipses with widely-scattered distributions of eccentricity and size. Physical Review E 62 (5), 5948–5952.
de Dreuzy, J.R., Davy, P., Bour, O., 2001b. Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 1-effective connectivity. Water Resources Research 37, 8.
de Dreuzy, J.R., Davy, P., Bour, O., 2001c. Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 2-Permeability of networks based on log-normal distribution of apertures. Water Resources Research 37 (8), 2079–2095.
de Dreuzy, J.R., Davy, P., Bour, O., 2002. Permeability of 2D fracture networks with power-law distributions of length and aperture. Water Resources Research 38, 12.
de Dreuzy, J.R., Erhel, J., 2002. Efficient algorithms for the determination of the connected fracture network and the solution of the steady-state flow equation in fracture networks. Computers and Geosciences 29, 107–111.
de Marsily, G., et al., 2005. Dealing with spatial heterogeneity. Hydrogeology Journal 13, 161–183.
Dershowitz, W.S., Einstein, H.H., 1988. Characterizing rock joint geometry with joint system models. Rock Mechanics and Rock Engineering 21 (1), 21–51.
Dershowitz, W.S., Fidelibus, C., 1999. Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method. Water Resources Research 35 (9), 2685–2691.
Dorn, C., Linde, N., Le Borgne, T., Bour, O., Baron, L., 2011. Single-hole GPR reflection imaging of solute transport in a granitic aquifer. Geophysical Research Letters 38, 5.
Doughty, C., Long, J.C.S., Hestir, K., Benson, S.M., 1994. Hydrologic characterization of heterogeneous geologic media with an inverse method based on iterated function systems. Water Resources Research 30 (6), 1721–1745.
Dverstop, B., Andersson, J., 1989. Application of the discrete fracture network concept with filed data: possibilities of model calibration and validation. Water Resources Research 25 (3), 540–550.
Erhel, J., de Dreuzy, J.-R., Bresciani, E., 2008. Multi-parametric intensive stochastic simulations for hydrogeology on a computational grid. In: Tromeur-Dervout, D., Brenner, J., Emerson, D., Erhel, J. (Eds.), Parallel Computational Fluid Dynamics. Springer, pp. 389–397, Lecture Notes in Computational Science and Engineering.
Erhel, J., de Dreuzy, J.-R., Poirriez, B., 2009a. Flow simulation in three-dimensional discrete fracture networks. SIAM Journal on Scientific Computing 31 (4), 2688–2705.
Erhel, J., de Dreuzy, J.R., Beaudoin, A., Bresciani, E., Tromeur-Dervout, D., 2009b. A parallel scientific software for heterogeneous hydrogeoloy. In: Tuncer, I.H., Gulcat, U., Emerson, D.R., Matsuno, K. (Eds.), Parallel Computational Fluid Dynamics 2007. Lecture Notes in Computational Science and Engineering. Springer, pp. 39–48.
Falgout, R.D., et al., 2005. Pursuing scalability for HYPRE's conceptual interfaces. ACM Transactions on Mathematical Software 31 (3).
Fourno, A., Greiner, C., Delay, F., Mouche, E., Benabderrahmane, H., 2004. Smeared fractures: a promising approach to model transfers in fractured media. In: Miller, C.T., Farthing, M.W., Gray, W.G., Pinder, G.F. (Eds.), Computational Methods in Water Resources, vols. 1 and 2. Elsevier Science Bv, Amsterdam, pp. 1003–1014, Developments in Water Science.
Frampton, A., Cvetkovic, V., 2007. Upscaling particle transport in discrete fracture networks: 1. Nonreactive tracers. Water Resources Research 43, 10.
Gylling, B., Birgersson, L., Moreno, L., Neretnieks, I., 1998. Analysis of a long-term pumping and tracer test using the channel network model. Journal of Contaminant Hydrology 32 (3–4), 203–222.
Hsieh, P.A., 1998. Scale Effects in Fluid Flow Through Fractured Geological Media, Scale Dependence and Scale Invariance in Hydrology. Cambridge University Press 335-353.
Kalbacher, T., et al., 2007. Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the Grimsel rock laboratory. Computational Geosciences 11 (1), 9–26.
Kirkpatrick, S., 1971. Classical transport in disordered media: scaling and effective-medium theories. Physical Review Letters 27 (25), 1722–1725.
Kozlov, B., Lagues, M., 2010. Universality of 3D percolation exponents and first-order corrections to scaling for conductivity exponents. Physica a-Statistical Mechanics and its Applications 389 (23), 5339–5346.
L'Ecuyer, P., Simard, R., Chen, E.J., Kelton, W.D., 2002. An object-oriented random-number package with many long streams and substreams. Operations Research 50 (6), 1073–1075.
Le Goc, R., de Dreuzy, J.R., Davy, P., 2009. Statistical characteristics of flow as indicators of channeling in heterogeneous porous and fractured media. Advances in Water Resources 33 (3), 257–269.
Le Goc, R., de Dreuzy, J.R., Davy, P., 2010. Inverse problem strategy to identify flow channels in fractured media. Advances in Water Resources 33 (7), 782–800.
Lenti, V., Fidelibus, C., 2003. A BEM solution of steady-state flow problems in discrete fracture networks with minimization of core storage. Computers & Geosciences 29 (9), 1183–1190.
Long, J.C.S., Gilmour, P., Witherspoon, P.A., 1985. A model for steady fluid flow in random three-dimensional networks of disc-shaped fractures. Water Resources Research 21 (8), 1105–1115.
Long, J.C.S., Remer, J.S., Wilson, C.R., Witherspoon, P.A., 1982. Porous media equivalents for networks of discontinuous fractures. Water Resources Research 18 (3), 645–658.
Maryka, J., Severýn, O., Vohralík, M., 2004. Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model. Computational Geosciences 8, 3.
Méheust, Y., Schmittbuhl, J., 2001. Geometrical heterogeneities and permeability anisotropy of rough fractures. Journal of Geophysical Research-Solid Earth 106 (B2), 2089–2102.
Méheust, Y., Schmittbuhl, J., 2003. Scale effects related to flow in rough fractures. Pure and Applied Geophysics 160 (5-6), 1023–1050.
Mustapha, H., 2005. Simulation Numérique de L'écoulement Dans des Milieux Fracturés Tridimensionnels. University of Rennes 1.
National Research Council, 1996. Rock Fractures and Fluid Flow. National Academy Press, Washington, D.C..
Neuman, S.P., 2005. Trends, prospects and challenges in quantifying flow and transport through fractured rocks. Hydrogeology Journal 13 (1), 124–147.
Noetinger, B., Jarrige, N., 2012. A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks. Journal of Computational Physics 231 (1), 23–38.
Nordqvist, A.W., Tsang, Y.W., Tsang, C.F., Dverstop, B., Andersson, J., 1992. A variable aperture fracture network model for flow and transport in fractured rocks. Water Resources Research 28 (6), 1703–1713.
Normand, J.M., Herrmann, H.J., 1995. Precise determination of the conductivity exponent of 3D percolation using ''Percola''. International Journal of Modern Physics C-Physics and Computers 6 (6), 813–817.
Pichot, G., de Dreuzy, J.-R., Erhel, J., 2012. Flow simulation in 3D multi-scale fractured networks using non-matching meshes. SIAM Journal on Scientific Computing (SISC) 34, 1.
Pichot, G., Erhel, J., de Dreuzy, J.R., 2010. A mixed hybrid Mortar method for solving flow in discrete fracture networks. Applicable Analysis 89 (10), 1643–1729.
Poirriez, B., 2011. Étude et mise en oeuvre d'une méthode de sous-domaines pour la modélisation de l'écoulement dans des réseaux de fractures en 3D. University of Rennes 1.
Raviart, P.A., Thomas, J.M., 1977. A mixed finite element method for second order elliptic problems. In: Galligani, I., Magenes, E. (Eds.), Mathematical Aspects of the Finite Element Method. Springer Verlag, New York.
Renard, P., Allard, D., 2012. Connectivity metrics for subsurface flow and transport. Advances in Water Resources(0).
Roberts, J.-E., Thomas, J.-M., 1991. Mixed and hybrid methods. In: Ciarlet, P.G., Lions, J.L. (Eds.), Handbook of Numerical Analysis 2, Finite Element Methods - part 1. Elsevier Science Publishers B.V.. North-Holland, pp. 523–639.
Schmittbuhl, J., Gentier, S., Roux, S., 1993. Field-measurements of the roughness of fault surfaces. Geophysical Research Letters 20, 8.
Stauffer, D., Aharony, A., 1992. Introduction to Percolation Theory, second ed. Taylor and Francis, Bristol.
Sudicky, E.A., McLaren, R.G., 1992. The laplace transform galerkin technique for large-scale simulation of mass transport in discretly fractured porous formations. Water Resources Research 28 (2), 499–514.

Taylor, G., 1953. Dispersion of soluble matter in solvent flowing slowly through a tube. Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences 219 (1137), 186–203.

Tsang, C.-F., Neretnieks, I., 1998. Flow channeling in heterogeneous fractured rocks. Reviews of Geophysics 36 (2), 257–298.

Tsang, Y.W., Tsang, C.F., Hale, F.V., Dverstop, B., 1996. Tracer transport in a stochastic continuum model of fractured media. Water Resources Research 32 (10), 3077–3092.

Verscheure, M., Fourno, A., Chilès, J.-P., 2012. Joint inversion of fracture model properties for $CO_2$ storage monitoring or oil recovery history matching. Oil & Gas Science and Technology, Rev. IFP Energies nouvelles, Reservoir monitoring for $CO_2$ storage and oil & gas production.

Vohralik, M., Maryska, J., Severyn, O., 2007. Mixed and nonconforming finite element methods on a system of polygons. Applied Numerical Mathematics 57 (2), 176–193.

Warren, J.E., Root, P.J., Aime, M., 1963. The behavior of naturally fractured reservoirs. Society of Petroleum Engineers Journal, 245–255, September.

Wellman, T.P., Shapiro, A.M., Hill, M.C., 2009. Effects of simplifying fracture network representation on inert chemical migration in fracture-controlled aquifers. Water Resources Research 45, 21.

Xu, C.S., Dowd, P., 2009. A new computer code for discrete fracture network modelling. Computers & Geosciences 36 (3), 292–301.