Erratum to: Parallel vertex approximate gradient discretization of hybrid dimensional Darcy flow and transport in discrete fracture networks

Feng Xing, Roland Masson, Simon Lopez

Comput Geosci (2017) 21:595–617
DOI 10.1007/s10596-016-9606-z

ORIGINAL PAPER

Parallel vertex approximate gradient discretization of hybrid dimensional Darcy flow and transport in discrete fracture networks

Feng Xing[1,2] · **Roland Masson**[1] · **Simon Lopez**[2]

Received: 10 February 2016 / Accepted: 14 November 2016 / Published online: 13 December 2016

Abstract This paper proposes a parallel numerical algorithm to simulate the flow and the transport in a discrete fracture network taking into account the mass exchanges with the surrounding matrix. The discretization of the Darcy fluxes is based on the Vertex Approximate Gradient finite volume scheme adapted to polyhedral meshes and to heterogeneous anisotropic media, and the transport equation is discretized by a first-order upwind scheme combined with an Euler explicit integration in time. The parallelization is based on the single program, multiple data (SPMD) paradigm and relies on a distribution of the mesh on the processes with one layer of ghost cells in order to allow for a local assembly of the discrete systems. The linear system for the Darcy flow is solved using different linear solvers and preconditioners implemented in the PETSc and Trilinos libraries. The convergence of the scheme is validated on two original analytical solutions with one and four intersecting fractures. Then, the parallel efficiency of the algorithm is assessed on up to 512 processes with different types of meshes, different matrix fracture permeability ratios, and different levels of complexity of the fracture network.

Keywords Hybrid dimensional Darcy flow and transport · Fracture networks · Parallel Vertex Approximate Gradient (VAG)

The original version of this article was revised: modifications have been made in some inline and displayed equations. The full information regarding the corrections made can be found in the erratum for this article.

✉ Feng Xing
feng.xing@unice.fr

Roland Masson
roland.masson@unice.fr

Simon Lopez
s.lopez@brgm.fr

[1] Laboratoire de Mathématiques J.A. Dieudonné, UMR 7351 CNRS, University Nice Sophia Antipolis, team COFFEE, INRIA Sophia Antipolis Méditerranée, Parc Valrose 06108 Nice Cedex 02, France

[2] BRGM, Scientific and Technical Center, 3 Avenue Claude Guillemin, BP 36009, 45060 Orléans Cedex 2, France

1 Introduction

1.1 Hybrid dimensional flow and transport models

This article deals with the simulation of the Darcy flow and transport in fractured porous media for which the fractures are modeled as interfaces of codimension one. In this framework, the d-1 dimensional flow and transport in the fractures is coupled with the d dimensional flow and transport in the matrix leading to the so called hybrid dimensional Darcy flow and transport model.

For the Darcy flow model, we focus on the particular case where the pressure is continuous at the interfaces between the fractures and the matrix domain. This type of Darcy flow model introduced in [3, 4] corresponds physically to pervious fractures for which the ratio of the normal permeability of the fracture to the width of the fracture is large compared with the ratio of the permeability of the matrix to the size of the domain. Note that it does not cover the case of fractures acting as barriers for which the pressure is discontinuous at the matrix fracture interfaces (see [5, 22, 33] for discontinuous pressure models). It is also assumed in our model that the pressure is continuous at the fracture intersections. It corresponds to the assumption that the ratio between the

permeability at the fracture intersections and the width of the fracture is large compared to the ratio between the tangential permeability of each fracture and its length. We refer to [24] and [40] for more general reduced models taking into account discontinuous pressures at fracture intersections in dimension $d = 2$.

The hybrid dimensional transport model is derived in [3] in the case of a convection diffusion flux for the matrix and fracture concentration. In this work, a purely advective model is considered. It requires the specification of the transmission conditions at the matrix fracture interfaces and at fracture intersections which, to our knowledge, have not been done so far at the continuous level.

The discretization of the hybrid dimensional Darcy flow model with continuous pressures has been the object of several works. In [31], a cell-centered finite volume scheme using a two-point flux approximation (TPFA) is proposed assuming the orthogonality of the mesh and isotropic permeability fields. Cell-centered finite volume schemes can be extended to general meshes and anisotropic permeability fields using multipoint flux approximations (MPFA) following the ideas of [1, 2, 39, 42]. In [3] and [29], a mixed finite element (MFE) method is proposed, and control volume finite element methods (CVFE) using nodal unknowns have been introduced for such models in [9, 28, 34, 35, 38]. The hybrid finite volume and mimetic finite difference schemes, belonging to the family of hybrid mimetic mixed methods [17], have been extended to hybrid dimensional models in [6, 23] as well as in [12, 13] in the more general gradient discretization framework [18]. Non-matching discretizations of the fracture and matrix meshes are studied in [8, 16, 25] and [40].

Regarding the hybrid dimensional advective transport model, an explicit first-order upwind scheme combined with the MPFA Darcy fluxes is used in [1, 2] and [39]. At fracture intersections, the authors neglect the accumulation term and the concentration unknown is eliminated using the flux conservation equation in order to avoid severe restrictions on the time step caused by the small volumes. A CVFE method is used in [38] with a first-order upwind approximation and a fully implicit time integration of the two-phase flow model to avoid small time steps. Higher order methods have also been developed in the CVFE method of [34] using a MUSCL type second-order scheme for the saturation equation and also in [29] where a discontinuous Galerkin method is used for the transport saturation equation with an Euler implicit time integration in the fracture network and an explicit time integration in the matrix domain. In [26], a streamline method is developed in 2D based on the hybrid dimensional Darcy flow velocity field. The solution is very accurate for purely advective transport but this method requires that the fractures be expanded and seems difficult to extend to the case of a complex 3D network in practice.

1.2 Content and objectives of this work

In this work, we focus on the vertex approximate gradient (VAG) scheme introduced in [19] for diffusion problems and extended in [11–13] to hybrid dimensional Darcy flow models. The VAG scheme uses nodal and fracture-face unknowns in addition to the cell unknowns which can be eliminated without any fill-in. Thanks to its essentially nodal feature, it leads to a sparse discretization on tetrahedral or mainly tetrahedral meshes. The VAG scheme has the major advantage, compared with the CVFE methods of [9, 35, 38] or [34], that it avoids the mixing of the control volumes at the fracture matrix interfaces, which is a key feature for its coupling with the transport model. As shown in [11] for two-phase flow problems, the VAG scheme allows for a coarser mesh size at the matrix fracture interface for a given accuracy. For the discretization of the transport hybrid dimensional model, we will use in this work a simple first-order upwind scheme with explicit time integration. The extension to second-order MUSCL type discretization will be considered in a future work. Our main objective in this paper is to develop a parallel algorithm for the VAG discretization of hybrid dimensional Darcy flow and transport models, and to assess the parallel scalability of the algorithm.

Starting from the hybrid dimensional Darcy flow model of [11] and [12], we first derive the hybrid dimensional transport model for a general fracture network taking into account fracture intersections and the coupling with the matrix domain. Then, the VAG discretization of the Darcy flow model is recalled and the VAG Darcy fluxes are used to discretize the transport model with an upwind first-order discretization in space and an Euler explicit time integration. A key feature of this discretization is the definition of the control volumes which is adapted to the heterogeneities of the porous medium. This can be achieved thanks to the fact that, on the one hand, the VAG scheme keeps the cell unknowns and, on the other hand, the VAG Darcy fluxes are constructed independently of the definition of the control volumes. In particular, the control volumes are constructed in such a way that, at matrix fracture interfaces, the volume is taken only in the fracture. Otherwise, the fracture will be enlarged artificially and the front velocity will not be accurately approximated in the fractures as it it the case for usual CVFE methods. Note also that we do not eliminate the concentration unknowns at fracture intersections as was done in [1, 39] and [2] for cell-centered discretizations. In the case of a nodal discretization like the VAG scheme, this elimination is not possible since these unknowns are connected

to the matrix and it is not needed since the size of the control volumes at fracture intersections is roughly the same as the size of any control volume located at the matrix fracture interface.

Our parallelization of the hybrid dimensional flow and transport numerical model is based on the single program, multiple data (SPMD) paradigm. It relies on a distribution of the mesh on the processes with one layer of ghost cells in order to allow for a local assembly of the discrete systems. The linear system for the Darcy flow is solved using different linear solvers and preconditioners implemented in the PETSc and Trilinos libraries.

In order to validate the convergence of the scheme, two analytical solutions are constructed for the hybrid dimensional flow and transport model. We consider the case of a single non-immersed fracture as well as the case of four intersecting fractures. The analytical solutions for the transport model are obtained by integration of the matrix and fracture equations along the characteristics of the velocity field taking into account source terms coming from the matrix fracture transmission conditions. Then, we study the parallel scalability of the Darcy flow and transport solvers on up to 512 processes. Our numerical investigation includes different levels of complexity of the fracture network with a number of fractures ranging from a few to a few hundreds. It covers different types of meshes namely hexahedral, tetrahedral, and prismatic meshes as well as a large range of permeability ratios between the fracture network and the matrix domain. In addition, the influence of the choices of the linear solver and of the preconditioner is also studied for the solution of the Darcy flow equation.

The paper is organized as follows. Section 2 recalls the geometrical and functional setting introduced in [12] for a general 2D fracture network immersed in a surrounding 3D matrix. Then, the hybrid dimensional Darcy flow and transport models are introduced. In Section 3, the VAG discretization is recalled for the Darcy flow model and extended to the transport model. The parallel implementation of the scheme is detailed in Section 4. Section 5 is devoted to the description of the test cases including the analytical solutions and to the numerical investigation of the parallel scalability of the algorithm.

2 Hybrid dimensional Darcy flow and transport model in fractured porous media

2.1 Discrete fracture network and functional setting

Let Ω denote a bounded domain of $\mathbb{R}^d$, $d = 2, 3$ assumed to be polyhedral for $d = 3$ and polygonal for $d = 2$. To fix ideas the dimension will be fixed to $d = 3$ when it needs to be specified, for instance in the naming of the geometrical objects or for the space discretization in the next section. The adaptations to the case $d = 2$ are straightforward.

We consider the asymptotic model introduced in [3] where fractures are represented as interfaces of codimension 1. Let I be a finite set and let $\overline{\Gamma} = \bigcup_{i\in I} \overline{\Gamma}_i$ and its interior $\Gamma = \overline{\Gamma} \setminus \partial\overline{\Gamma}$ denote the network of fractures $\Gamma_i \subset \Omega$, $i \in I$, such that each Γ_i is a planar, polygonal, simply connected, open domain included in an oriented plane $\mathcal{P}_i$ of $\mathbb{R}^d$. It is assumed that the angles of Γ_i are strictly smaller than 2π and that $\Gamma_i \cap \overline{\Gamma}_j = \emptyset$ for all $i \neq j$. For all $i \in I$, let us set $\Sigma_i = \partial\Gamma_i$, $\Sigma_{i,j} = \Sigma_i \cap \Sigma_j$, $j \in I \setminus \{i\}$, $\Sigma_{i,0} = \Sigma_i \cap \partial\Omega$, $\Sigma_{i,N} = \Sigma_i \setminus (\bigcup_{j\in I\setminus\{i\}} \Sigma_{i,j} \cup \Sigma_{i,0})$, and $\Sigma = \bigcup_{(i,j)\in I\times I, i\neq j} \Sigma_{i,j}$. It is assumed that $\Sigma_{i,0} = \overline{\Gamma}_i \cap \partial\Omega$ (Fig. 1).

We will denote by $d\tau(\mathbf{x})$ the $d-1$ dimensional Lebesgue measure on Γ. On the fracture network Γ, we define the function space $L^2(\Gamma) = \{v = (v_i)_{i\in I}, v_i \in L^2(\Gamma_i), i \in I\}$, endowed with the norm $\|v\|^2_{L^2(\Gamma)} = \sum_{i\in I} \|v_i\|^2_{L^2(\Gamma_i)}$. Its subspace $H^1(\Gamma)$ is defined as the space of functions $v = (v_i)_{i\in I}$ such that $v_i \in H^1(\Gamma_i)$, $i \in I$ with continuous traces at the fracture intersections. The space $H^1(\Gamma)$ is endowed with the norm $\|v\|^2_{H^1(\Gamma)} = \sum_{i\in I} \|v_i\|^2_{H^1(\Gamma_i)}$ and its subspace with vanishing traces on $\Sigma_0 = \bigcup_{i\in I} \Sigma_{i,0}$ is denoted by $H^1_{\Sigma_0}(\Gamma)$.

Let us also consider the trace operator γ_i from $H^1(\Omega)$ to $L^2(\Gamma_i)$ as well as the trace operator γ from $H^1(\Omega)$ to $L^2(\Gamma)$ such that $(\gamma v)_i = \gamma_i(v)$ for all $i \in I$.

On Ω, the gradient operator from $H^1(\Omega)$ to $L^2(\Omega)^d$ is denoted by ∇. On the fracture network Γ, the tangential gradient ∇_τ acting from $H^1(\Gamma)$ to $L^2(\Gamma)^{d-1}$ is defined by

$$\nabla_\tau v = (\nabla_{\tau_i} v_i)_{i\in I},$$

where, for each $i \in I$, the tangential gradient ∇_{τ_i} is defined from $H^1(\Gamma_i)$ to $L^2(\Gamma_i)^{d-1}$ by fixing a reference Cartesian coordinate system of the plane $\mathcal{P}_i$ containing Γ_i. We also denote by div_{τ_i} the divergence operator from $H_{\mathrm{div}}(\Gamma_i)$ to $L^2(\Gamma_i)$.

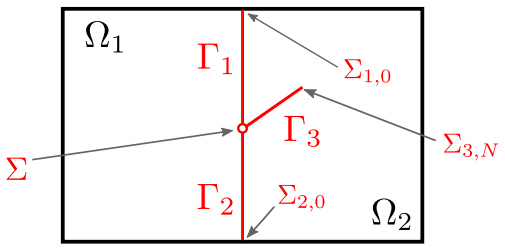


Fig. 1 Example of a 2D domain with three intersecting fractures $\Gamma_1, \Gamma_2, \Gamma_3$ and two connected components Ω_1, Ω_2

The function spaces arising in the variational formulation of the hybrid dimensional Darcy flow model are

$$V = \{v \in H^1(\Omega) \text{ such that} \gamma v \in H^1(\Gamma)\},$$

and its subspace

$$V^0 = \{v \in H^1_0(\Omega) \text{ such that} \gamma v \in H^1_{\Sigma_0}(\Gamma)\}.$$

The space V^0 is endowed with the following Hilbertian norm

$$\|v\|_{V^0} = \left(\|\nabla v\|^2_{L^2(\Omega)^d} + \|\nabla_\tau \gamma v\|^2_{L^2(\Gamma)^{d-1}}\right)^{1/2}.$$

Let $\Omega_\alpha, \alpha \in \mathcal{A}$ denote the connected components of $\Omega \setminus \overline{\Gamma}$, with $\mathcal{A}$ being the set of connected components of $\Omega \setminus \overline{\Gamma}$. Let us define the space $H_{\text{div}}(\Omega \setminus \overline{\Gamma}) = \{\mathbf{q}_m = (\mathbf{q}_{m,\alpha})_{\alpha\in\mathcal{A}} \mid \mathbf{q}_{m,\alpha} \in H_{\text{div}}(\Omega_\alpha)\}$. Using the orientation of $\mathcal{P}_i$, we can define the two sides $\pm$ of the fracture Γ_i, for all $i \in I$. For all $\mathbf{q}_m \in H_{\text{div}}(\Omega \setminus \overline{\Gamma})$, let $\gamma^{\pm}_{\mathbf{n},i}\mathbf{q}_m$ denote the normal trace of $\mathbf{q}_m$ on the side $\pm$ of Γ_i with the normal oriented outward from the side $\pm$. Let us define the Hilbert function space

$$\begin{aligned} H(\Omega, \Gamma) = \{&\mathbf{q}_m = (\mathbf{q}_{m,\alpha})_{\alpha\in\mathcal{A}}, \mathbf{q}_f = (\mathbf{q}_{f,i})_{i\in I} \mid \\ &\mathbf{q}_m \in H_{\text{div}}(\Omega \setminus \overline{\Gamma}),\ \mathbf{q}_{f,i} \in L^2(\Gamma_i)^{d-1}, \\ &\text{div}_{\tau_i}(\mathbf{q}_{f,i}) - \gamma^+_{\mathbf{n},i}\mathbf{q}_m - \gamma^-_{\mathbf{n},i}\mathbf{q}_m \in L^2(\Gamma_i), i \in I\}, \end{aligned}$$

and its closed Hilbert subspace

$$\begin{aligned} W(\Omega, \Gamma) = \Bigg\{&(\mathbf{q}_m, \mathbf{q}_f) \in H(\Omega, \Gamma) \mid \sum_{\alpha\in\mathcal{A}} \int_{\Omega_\alpha} (\mathbf{q}_{m,\alpha} \cdot \nabla v \\ &+ \text{div}(\mathbf{q}_{m,\alpha}) v) d\mathbf{x} + \sum_{i\in I} \int_{\Gamma_i} (\mathbf{q}_{f,i} \cdot \nabla_{\tau_i} \gamma_i v \\ &+ (\text{div}_{\tau_i}(\mathbf{q}_{f,i}) - \gamma^+_{\mathbf{n},i}\mathbf{q}_m - \gamma^-_{\mathbf{n},i}\mathbf{q}_m)\gamma_i v) d\tau(\mathbf{x}) \\ &= 0 \,\forall\, v \in V^0 \Bigg\}. \end{aligned} \tag{1}$$

The last definition corresponds to imposing in a weak sense the conditions $\sum_{i\in I} \gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i} = 0$ on $\Sigma \setminus \Sigma_0$ and $\gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i} = 0$ on $\Sigma_{i,N}, i \in I$, where $\gamma_{\mathbf{n},\Sigma_i}$ is the normal trace operator on Σ_i (tangent to Γ_i) with the normal oriented outward from Γ_i, and using the extension of $\gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i}$ by zero on $\Sigma \setminus \Sigma_i$.

2.2 Hybrid dimensional Darcy flow model

In the matrix domain $\Omega \setminus \overline{\Gamma}$ (resp. in the fracture network Γ), let us denote by $\Lambda_m \in L^\infty(\Omega)^{d\times d}$ (resp. $\Lambda_f \in L^\infty(\Gamma)^{(d-1)\times(d-1)}$) the permeability tensor such that there exist $\overline{\lambda}_m \geq \underline{\lambda}_m > 0$ (resp. $\overline{\lambda}_f \geq \underline{\lambda}_f > 0$) with

$$\underline{\lambda}_m|\boldsymbol{\xi}|^2 \leq (\Lambda_m(\mathbf{x})\boldsymbol{\xi}, \boldsymbol{\xi}) \leq \overline{\lambda}_m|\boldsymbol{\xi}|^2 \text{ for all} \boldsymbol{\xi} \in \mathbb{R}^d, \mathbf{x} \in \Omega,$$

$$(\text{resp. } \underline{\lambda}_f|\boldsymbol{\xi}|^2 \leq (\Lambda_f(\mathbf{x})\boldsymbol{\xi}, \boldsymbol{\xi}) \leq \overline{\lambda}_f|\boldsymbol{\xi}|^2 \text{ for all} \boldsymbol{\xi} \in \mathbb{R}^{d-1}, \mathbf{x} \in \Gamma).$$

We also denote by μ the fluid viscosity which is assumed constant and by $d_f \in L^\infty(\Gamma)$ the width of the fractures assumed to be such that there exist $\overline{d}_f \geq \underline{d}_f > 0$ with $\underline{d}_f \leq d_f(\mathbf{x}) \leq \overline{d}_f$ for all $\mathbf{x} \in \Gamma$.

Given $\bar{u} \in V$, the strong formulation of the hybrid dimensional Darcy flow model amounts to: find $u \in V$ and $(\mathbf{q}_m, \mathbf{q}_f) \in W(\Omega, \Gamma)$ such that $u - \bar{u} \in V^0$ and

$$\begin{cases} \text{div}(\mathbf{q}_{m,\alpha}) = 0 & \text{on } \Omega_\alpha, \alpha \in \mathcal{A}, \\ \mathbf{q}_{m,\alpha} = -\frac{\Lambda_m}{\mu}\nabla u & \text{on } \Omega_\alpha, \alpha \in \mathcal{A}, \\ \text{div}_{\tau_i}(\mathbf{q}_{f,i}) - \gamma^+_{\mathbf{n},i}\mathbf{q}_m - \gamma^-_{\mathbf{n},i}\mathbf{q}_m = 0 & \text{on } \Gamma_i, i \in I, \\ \mathbf{q}_{f,i} = -d_f\frac{\Lambda_f}{\mu}\nabla_{\tau_i}\gamma_i u & \text{on } \Gamma_i, i \in I. \end{cases} \tag{2}$$

The weak formulation of Eq. 2 amounts to: find $u \in V$ such that $u - \bar{u} \in V^0$ and the following variational equation is satisfied for all $v \in V^0$:

$$\begin{aligned} &\int_\Omega \frac{\Lambda_m(\mathbf{x})}{\mu}\nabla u(\mathbf{x}) \cdot \nabla v(\mathbf{x}) d\mathbf{x} \\ &\quad + \int_\Gamma d_f(\mathbf{x})\frac{\Lambda_f(\mathbf{x})}{\mu}\nabla_\tau \gamma u(\mathbf{x}) \cdot \nabla_\tau \gamma v(\mathbf{x}) d\tau(\mathbf{x}) = 0. \end{aligned} \tag{3}$$

The existence and uniqueness of the solution to Eq. 3 derives from the Lax Milgram theorem and a Poincaré inequality stated in [12].

2.3 Hybrid dimensional transport model

Let $\gamma_{\mathbf{n}}$ be the normal trace operator on $\partial\Omega$ with the normal oriented outward from Ω. Let us define $\partial\Omega^- = \{\mathbf{x} \in \partial\Omega \mid \gamma_{\mathbf{n}}\mathbf{q}_m(\mathbf{x}) < 0\}$, $\Sigma^-_{i,0} = \{\mathbf{x} \in \Sigma_{i,0} \mid \gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i}(\mathbf{x}) < 0\}$, $i \in I$, as well as the following subset of $\Sigma \setminus \Sigma_0$:

$$\Sigma^- = \{\mathbf{x} \in \Sigma \setminus \Sigma_0 \mid \sum_{i\in I} |\gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i}(\mathbf{x})| \neq 0\}.$$

We consider a linear, purely advective model with velocity $\mathbf{q}_m$ in the matrix domain and $\mathbf{q}_f$ in the fracture network. The matrix concentration is denoted by c_m ($c_{m,\alpha}$ in each connected component Ω_α, $\alpha \in \mathcal{A}$) and the fracture concentration, representing the average concentration in the fracture width, is denoted by c_f ($c_{f,i}$ in each fracture Γ_i, $i \in I$). The 2D equation in the fracture network is as usual obtained by integration of the 3D advection equation in the width of the fractures. For a purely advective equation, the transmission condition at the matrix fracture interfaces states that the input normal flux in the matrix is obtained using the upwind fracture concentration c_f. At the fracture intersection Σ^-, an additional unknown c_{f,Σ^-} must be introduced and the transmission conditions state that the normal fluxes sum to zero and that the input normal fluxes are obtained using the upwind concentration c_{f,Σ^-}.

Let be given the input boundary conditions $\bar{c}_m \in L^\infty(\partial\Omega^-)$, $\bar{c}_{f,i} \in L^\infty(\Sigma_{i,0}^-)$, $i \in I$, and the initial conditions $c_m^0 \in L^\infty(\Omega \setminus \overline{\Gamma})$, $c_f^0 \in L^\infty(\Gamma)$. Let us denote by $\phi_m(\mathbf{x})$ the porosity in the matrix and by $\phi_f(\mathbf{x})$ the porosity in the fracture network. The transport hybrid dimensional model amounts to: find $c_m \in L^\infty\left((\Omega \setminus \overline{\Gamma}) \times (0,T)\right)$, $c_f \in L^\infty(\Gamma \times (0,T))$, and $c_{f,\Sigma^-} \in L^\infty\left(\Sigma^- \times (0,T)\right)$, such that:

$$
\begin{cases}
\phi_m \partial_t c_{m,\alpha} + \operatorname{div}(c_{m,\alpha}\mathbf{q}_{m,\alpha}) = 0 & \text{on } \Omega_\alpha \times (0,T), \alpha \in \mathcal{A} \\
\phi_f d_f \partial_t c_{f,i} + \operatorname{div}_{\tau_i}(c_{f,i}\mathbf{q}_{f,i}) = \gamma_{\mathbf{n},i}^+ c_m \mathbf{q}_m + \gamma_{\mathbf{n},i}^- c_m \mathbf{q}_m & \text{on } \Gamma_i \times (0,T), i \in I, \\
(\gamma_{\mathbf{n},i}^\pm c_m \mathbf{q}_m)^- = c_f(\gamma_{\mathbf{n},i}^\pm \mathbf{q}_m)^- & \text{on } \Gamma_i \times (0,T), i \in I, \\
(\gamma_{\mathbf{n},\Sigma_i} c_{f,i}\mathbf{q}_{f,i})^- = c_{f,\Sigma^-}(\gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i})^- & \text{on } (\Sigma_i \setminus \Sigma_{i,0}) \times (0,T), i \in I, \\
\sum_{j\in I} \gamma_{\mathbf{n},\Sigma_j} c_{f,j}\mathbf{q}_{f,j} = 0 & \text{on } (\Sigma \setminus \Sigma_0) \times (0,T), \\
(\gamma_{\mathbf{n}} c_m \mathbf{q}_m)^- = \bar{c}_m(\gamma_{\mathbf{n}} q_m)^- & \text{on } \partial\Omega \times (0,T), \\
(\gamma_{\mathbf{n},\Sigma_i} c_{f,i}\mathbf{q}_{f,i})^- = \bar{c}_{f,i}(\gamma_{\mathbf{n},\Sigma_i}\mathbf{q}_{f,i})^- & \text{on } \Sigma_{i,0} \times (0,T), i \in I, \\
c_m = c_m^0 & \text{on } (\Omega \setminus \overline{\Gamma}) \times \{t=0\}, \\
c_f = c_f^0 & \text{on } \Gamma \times \{t=0\},
\end{cases}
\tag{4}
$$

where the notations $a^+ = \max(a,0)$ and $a^- = \min(a,0)$ are used for all $a \in \mathbb{R}$.

3 Vertex approximate gradient discretization

3.1 VAG discretization of the Darcy flow model

In the spirit of [19], we consider generalized polyhedral meshes of Ω in the sense that the edges of the mesh are linear but its faces are not necessarily planar. Roughly speaking, each face is assumed to be defined by the union of the triangles joining each edge of the face to a so called face center. This definition has the advantage to include in particular hexahedral cells with non planar faces.

Let $\mathcal{M}$ be the set of cells which are disjoint open polyhedral subsets of Ω such that $\bigcup_{K\in\mathcal{M}} \overline{K} = \overline{\Omega}$. For each $K \in \mathcal{M}$, it is assumed that there exists $\mathbf{x}_K \in K \setminus \partial K$, the so-called "center" of the cell K, such that K is star-shaped with respect to $\mathbf{x}_K$. We then denote by $\mathcal{F}_K$ the set of interfaces of non zero $d-1$ dimensional measure among the interior faces $\overline{K} \cap \overline{L}$, $L \in \mathcal{M} \setminus \{K\}$, and the boundary interface $\overline{K} \cap \partial\Omega$, which possibly splits in several boundary faces. Let us denote by

$$\mathcal{F} = \bigcup_{K\in\mathcal{M}} \mathcal{F}_K$$

the set of all faces of the mesh. Note that the faces are not assumed to be planar, hence the term "generalized polyhedral mesh". For $\sigma \in \mathcal{F}$, let $\mathcal{E}_\sigma$ be the set of interfaces of non zero, $d-2$ dimensional measure among the interfaces $\overline{\sigma} \cap \overline{\sigma}'$, $\sigma' \in \mathcal{F} \setminus \{\sigma\}$. Then, we denote by

$$\mathcal{E} = \bigcup_{\sigma\in\mathcal{F}} \mathcal{E}_\sigma$$

the set of all edges of the mesh. Let $\mathcal{V}_\sigma = \bigcup_{(e,e')\in\mathcal{E}_\sigma^2, e\neq e'} \left(e \cap e'\right)$ be the set of nodes of σ. For each $K \in \mathcal{M}$ we define $\mathcal{V}_K = \bigcup_{\sigma\in\mathcal{F}_K} \mathcal{V}_\sigma$, and we also denote by

$$\mathcal{V} = \bigcup_{K\in\mathcal{M}} \mathcal{V}_K$$

the set of all nodes of the mesh. It is then assumed that for each face $\sigma \in \mathcal{F}$, there exists a so-called "centre" of the face $\mathbf{x}_\sigma \in \sigma \setminus \bigcup_{e\in\mathcal{E}_\sigma} e$ such that $\mathbf{x}_\sigma = \sum_{\mathbf{s}\in\mathcal{V}_\sigma} \beta_{\sigma,\mathbf{s}}\mathbf{x}_\mathbf{s}$, with $\sum_{\mathbf{s}\in\mathcal{V}_\sigma} \beta_{\sigma,\mathbf{s}} = 1$, and $\beta_{\sigma,\mathbf{s}} \geq 0$ for all $\mathbf{s} \in \mathcal{V}_\sigma$; moreover, the face σ is assumed to be defined by the union of the triangles $T_{\sigma,e}$ defined by the face center $\mathbf{x}_\sigma$ and each edge $e \in \mathcal{E}_\sigma$.

The mesh is also supposed to be conforming w.r.t. the fracture network Γ in the sense that for each $i \in I$ there exists a subset $\mathcal{F}_{\Gamma_i}$ of $\mathcal{F}$ such that $\overline{\Gamma}_i = \bigcup_{\sigma\in\mathcal{F}_{\Gamma_i}} \overline{\sigma}$. We will denote by $\mathcal{F}_\Gamma$ the set of fracture faces $\bigcup_{i\in I} \mathcal{F}_{\Gamma_i}$. The following notations will be used for convenience:

$$
\begin{aligned}
\mathcal{M}_\mathbf{s} &= \{K \in \mathcal{M} \,|\, \mathbf{s} \in \mathcal{V}_K\}, \\
\mathcal{M}_\sigma &= \{K \in \mathcal{M} \,|\, \sigma \in \mathcal{F}_K\}, \\
\mathcal{F}_{\Gamma,\mathbf{s}} &= \{\sigma \in \mathcal{F}_\Gamma \,|\, \mathbf{s} \in \mathcal{V}_\sigma\},
\end{aligned}
$$

and

$$\mathcal{F}_{\Gamma,K} = \mathcal{F}_K \cap \mathcal{F}_\Gamma.$$

This geometrical discretization of Ω and Γ is denoted in the following by $\mathcal{D}$.

The VAG discretization was introduced in [19] for diffusive problems on heterogeneous, anisotropic media. Its extension to the hybrid dimensional Darcy model is based on the following vector space of unknowns:

$$X_\mathcal{D} = \{v_K, v_\mathbf{s}, v_\sigma \in \mathbb{R}, K \in \mathcal{M}, \mathbf{s} \in \mathcal{V}, \sigma \in \mathcal{F}_\Gamma\},$$

and its subspace with homogeneous Dirichlet boundary conditions on $\partial\Omega$:

$$X_{\mathcal{D}}^0 = \{v \in X_{\mathcal{D}} \mid v_{\mathbf{s}} = 0 \text{ for } \mathbf{s} \in \mathcal{V}_{ext}\}.$$

where $\mathcal{V}_{ext} = \mathcal{V} \cap \partial\Omega$ denotes the set of boundary nodes, and $\mathcal{V}_{int} = \mathcal{V} \setminus \partial\Omega$ denotes the set of interior nodes.

A finite element discretization of V is built using a tetrahedral sub-mesh of $\mathcal{M}$ and a second-order interpolation at the face centers $\mathbf{x}_\sigma$, $\sigma \in \mathcal{F} \setminus \mathcal{F}_\Gamma$ defined by the operator $I_\sigma : X_{\mathcal{D}} \to \mathbb{R}$ such that

$$I_\sigma(v) = \sum_{\mathbf{s} \in \mathcal{V}_\sigma} \beta_{\sigma,\mathbf{s}} v_{\mathbf{s}}.$$

The tetrahedral sub-mesh is defined by $\mathcal{T} = \{T_{K,\sigma,e}, e \in \mathcal{E}_\sigma, \sigma \in \mathcal{F}_K, K \in \mathcal{M}\}$ where $T_{K,\sigma,e}$ is the tetrahedron joining the cell center $\mathbf{x}_K$ to the triangle $T_{\sigma,e}$ (see Fig. 2 for examples of such tetrahedra).

For a given $v \in X_{\mathcal{D}}$, we define the function $\pi_{\mathcal{T}} v \in V$ as the continuous piecewise affine function on each tetrahedron of $\mathcal{T}$ such that $\pi_{\mathcal{T}} v(\mathbf{x}_K) = v_K$, $\pi_{\mathcal{T}} v(\mathbf{s}) = v_{\mathbf{s}}$, $\pi_{\mathcal{T}} v(\mathbf{x}_\sigma) = v_\sigma$, and $\pi_{\mathcal{T}} v(\mathbf{x}_{\sigma'}) = I_{\sigma'}(v)$ for all $K \in \mathcal{M}$, $\mathbf{s} \in \mathcal{V}$, $\sigma \in \mathcal{F}_\Gamma$, and $\sigma' \in \mathcal{F} \setminus \mathcal{F}_\Gamma$. The nodal basis of this finite element discretization will be denoted by η_K, $\eta_{\mathbf{s}}$, η_σ, for $K \in \mathcal{M}$, $\mathbf{s} \in \mathcal{V}$, $\sigma \in \mathcal{F}_\Gamma$.

The VAG discretization of the hybrid dimensional Darcy flow model (2) is based on its weak formulation (3). Given $\bar{u}_{\mathbf{s}}$, $\mathbf{s} \in \mathcal{V}_{ext}$, it amounts to: find $u_{\mathcal{D}} \in X_{\mathcal{D}}$ with $u_{\mathbf{s}} = \bar{u}_{\mathbf{s}}$ for all $\mathbf{s} \in \mathcal{V}_{ext}$ and such that for all $v_{\mathcal{D}} \in X_{\mathcal{D}}^0$ one has

$$\int_\Omega \frac{\Lambda_m(\mathbf{x})}{\mu} \nabla \pi_{\mathcal{T}} u_{\mathcal{D}}(\mathbf{x}) \cdot \nabla \pi_{\mathcal{T}} v_{\mathcal{D}}(\mathbf{x}) d\mathbf{x} + \int_\Gamma d_f(\mathbf{x}) \frac{\Lambda_f(\mathbf{x})}{\mu} \nabla_\tau \gamma \pi_{\mathcal{T}} u_{\mathcal{D}}(\mathbf{x}) \cdot \nabla_\tau \gamma \pi_{\mathcal{T}} v_{\mathcal{D}}(\mathbf{x}) d\tau(\mathbf{x}) = 0. \tag{5}$$

Following [12], this Galerkin finite element formulation (5) can be reformulated in terms of discrete conservation laws using the following definition of the VAG fluxes. For all $v_{\mathcal{D}} \in X_{\mathcal{D}}$, the VAG matrix fluxes connect the cell $K \in \mathcal{M}$ to its nodes and fracture faces $\nu \in \mathcal{V}_K \cup \mathcal{F}_{\Gamma,K}$:

$$\begin{aligned} F_{K,\nu}(v_{\mathcal{D}}) &= -\int_K \frac{\Lambda_m(\mathbf{x})}{\mu} \nabla \pi_{\mathcal{T}} v_{\mathcal{D}}(\mathbf{x}) \cdot \nabla \eta_\nu(\mathbf{x}) d\mathbf{x} \\ &= \sum_{\nu' \in \mathcal{V}_K \cup \mathcal{F}_{\Gamma,K}} a_{K,\nu}^{\nu'} (v_K - v_{\nu'}) \end{aligned} \tag{6}$$

with $a_{K,\nu}^{\nu'} = \int_K \frac{\Lambda_m(\mathbf{x})}{\mu} \nabla \eta_\nu(\mathbf{x}) \cdot \nabla \eta_{\nu'}(\mathbf{x}) d\mathbf{x}$. The VAG fracture fluxes connect the face $\sigma \in \mathcal{F}_\Gamma$ to its nodes $\mathbf{s} \in \mathcal{V}_\sigma$:

$$\begin{aligned} F_{\sigma,\mathbf{s}}(v_{\mathcal{D}}) &= -\int_\sigma d_f(\mathbf{x}) \frac{\Lambda_f(\mathbf{x})}{\mu} \nabla_\tau \gamma \pi_{\mathcal{T}} v_{\mathcal{D}}(\mathbf{x}) \cdot \nabla_\tau \gamma \eta_{\mathbf{s}}(\mathbf{x}) d\tau(\mathbf{x}) \\ &= \sum_{\mathbf{s}' \in \mathcal{V}_\sigma} a_{\sigma,\mathbf{s}}^{\mathbf{s}'} (v_\sigma - v_{\mathbf{s}'}) \end{aligned} \tag{7}$$

with $a_{\sigma,\mathbf{s}}^{\mathbf{s}'} = \int_\sigma d_f(\mathbf{x}) \frac{\Lambda_f(\mathbf{x})}{\mu} \nabla_\tau \gamma \eta_{\mathbf{s}}(\mathbf{x}) \cdot \nabla_\tau \gamma \eta_{\mathbf{s}'}(\mathbf{x}) d\tau(\mathbf{x})$ (Fig. 3).

Then, the Galerkin finite element formulation (5) is equivalent to: find $u_{\mathcal{D}} \in X_{\mathcal{D}}$ satisfying the following set

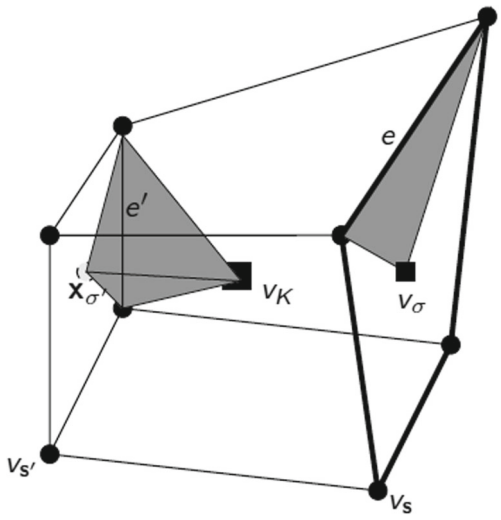


Fig. 2 For a cell K with one fracture face σ in bold: cell unknown v_K located at $\mathbf{x}_K$, fracture-face unknown v_σ located at $\mathbf{x}_\sigma$, node unknowns $v_{\mathbf{s}}$, $v_{\mathbf{s}'}$, face center $\mathbf{x}_{\sigma'}$ of face σ', triangle $T_{\sigma,e}$ (convex hull of e and $\mathbf{x}_\sigma$), triangle $T_{\sigma',e'}$ (convex hull of e' and $\mathbf{x}_{\sigma'}$) and tetrahedron $T_{K,\sigma',e'}$ (convex hull of $\mathbf{x}_K$, $\mathbf{x}_{\sigma'}$ and e')

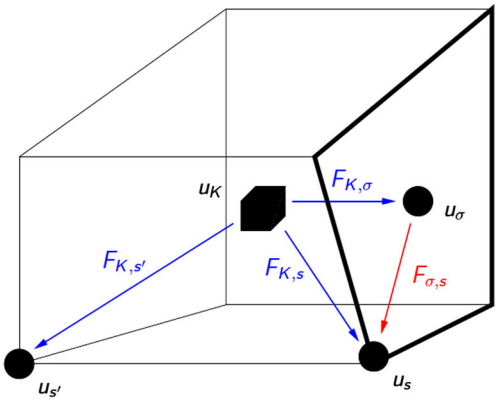


Fig. 3 Matrix fluxes (*in blue*) and fracture fluxes (*in red*) inside a cell K with a fracture face σ (*in bold*). The matrix fluxes $F_{K,\nu}$ connect the cell K to its nodes and fracture faces $\nu \in \mathcal{V}_K \cup \mathcal{F}_{\Gamma,K}$. The fracture fluxes $F_{\sigma,\mathbf{s}}$ connect the face σ to the nodes $\mathbf{s} \in \mathcal{V}_\sigma$ of σ

of discrete conservation equations and Dirichlet boundary conditions:

$$\begin{cases} \sum_{\mathbf{s}\in\mathcal{V}_K} F_{K,\mathbf{s}}(u_\mathcal{D}) + \sum_{\sigma\in\mathcal{F}_{\Gamma,K}} F_{K,\sigma}(u_\mathcal{D}) = 0, \ K\in\mathcal{M} \\ \sum_{\mathbf{s}\in\mathcal{V}_\sigma} F_{\sigma,\mathbf{s}}(u_\mathcal{D}) + \sum_{K\in\mathcal{M}_\sigma} -F_{K,\sigma}(u_\mathcal{D}) = 0, \ \sigma\in\mathcal{F}_\Gamma \\ \sum_{K\in\mathcal{M}_\mathbf{s}} -F_{K,\mathbf{s}}(u_\mathcal{D}) + \sum_{\sigma\in\mathcal{F}_{\Gamma,\mathbf{s}}} -F_{\sigma,\mathbf{s}}(u_\mathcal{D}) = 0, \ \mathbf{s}\in\mathcal{V}_{int}, \\ u_\mathbf{s} = \bar{u}_\mathbf{s}, \ \mathbf{s}\in\mathcal{V}_{ext}. \end{cases}$$

3.2 First-order upwind discretization of the transport model

3.2.1 Definition of control volumes

The VAG discretization of the hybrid dimensional transport model combines the VAG matrix and fracture fluxes (6), (7) with the following definition of the control volumes based on partitions of the cells and of the fracture faces. These partitions are respectively denoted, for all $K\in\mathcal{M}$, by

$$\overline{K} = \overline{\omega}_K \cup \left(\bigcup_{\mathbf{s}\in\mathcal{V}_K\setminus(\mathcal{V}_{ext}\cup\mathcal{V}_\Gamma)} \overline{\omega}_{K,\mathbf{s}} \right)$$

and, for all $\sigma\in\mathcal{F}_\Gamma$, by

$$\overline{\sigma} = \overline{\omega}_\sigma \cup \left(\bigcup_{\mathbf{s}\in\mathcal{V}_\sigma\setminus\mathcal{V}_{ext}} \overline{\omega}_{\sigma,\mathbf{s}} \right).$$

Then, the control volumes are defined by ω_K for all cells $K\in\mathcal{M}$, by ω_σ for all fracture faces $\sigma\in\mathcal{F}_\Gamma$, and by

$$\overline{\omega}_\mathbf{s} = \bigcup_{K\in\mathcal{M}_\mathbf{s}} \overline{\omega}_{K,\mathbf{s}},$$

for all nodes $\mathbf{s}\in\mathcal{V}_{int}\setminus\mathcal{V}_\Gamma$, and by

$$\overline{\omega}_\mathbf{s} = \bigcup_{\sigma\in\mathcal{F}_{\Gamma,\mathbf{s}}} \overline{\omega}_{\sigma,\mathbf{s}},$$

for all nodes $\mathbf{s}\in\mathcal{V}_\Gamma\setminus\mathcal{V}_{ext}$. Note that this definition avoid the mixing of the fracture and matrix rocktypes at the control volumes $\mathbf{s}\in\mathcal{V}_\Gamma\setminus\mathcal{V}_{ext}$ and $\sigma\in\mathcal{F}_\Gamma$. This is exhibited in Fig. 4 in comparison with an alternative choice mixing the matrix and fracture rocktypes which artificially enlarges the fractures. We refer to [12] for numerical comparisons on a two phase flow model of these two types of choices of the control volumes.

The same idea is applied for all nodes located at different rocktype interfaces. In practice, for such a node $\mathbf{s}\in\mathcal{V}_{int}\setminus\mathcal{V}_\Gamma$ (resp. $\mathbf{s}\in\mathcal{V}_\Gamma\setminus\mathcal{V}_{ext}$), the set $\omega_{K,\mathbf{s}}$ (resp. $\omega_{\sigma,\mathbf{s}}$) should be non empty only for the cell(s) K (resp. fracture face(s) σ) with the largest permeability among those around the node $\mathbf{s}$ (see [20] for details).

In practice, the above partitions of the cells and fracture faces does not need to be built. It is sufficient to define the matrix volume fractions

$$\alpha_{K,\mathbf{s}} = \frac{\int_{\omega_{K,\mathbf{s}}} d\mathbf{x}}{\int_K d\mathbf{x}}, \mathbf{s}\in\mathcal{V}_K\setminus(\mathcal{V}_{ext}\cup\mathcal{V}_\Gamma), K\in\mathcal{M},$$

constrained to satisfy $\alpha_{K,\mathbf{s}}\geq 0$, and $\sum_{\mathbf{s}\in\mathcal{V}_K\setminus(\mathcal{V}_{ext}\cup\mathcal{V}_\Gamma)}\alpha_{K,\mathbf{s}}\leq 1$, as well as the fracture volume fractions

$$\alpha_{\sigma,\mathbf{s}} = \frac{\int_{\omega_{\sigma,\mathbf{s}}} d_f(\mathbf{x})d\tau(\mathbf{x})}{\int_\sigma d_f(\mathbf{x})d\tau(\mathbf{x})}, \mathbf{s}\in\mathcal{V}_\sigma\setminus\mathcal{V}_{ext}, \sigma\in\mathcal{F}_\Gamma,$$

such that $\alpha_{\sigma,\mathbf{s}}\geq 0$, and $\sum_{\mathbf{s}\in\mathcal{V}_\sigma\setminus\mathcal{V}_{ext}}\alpha_{\sigma,\mathbf{s}}\leq 1$. Then, the porous volumes of the control volumes are set to

$$\begin{aligned} \phi_K &= \left(1 - \sum_{\mathbf{s}\in\mathcal{V}_K\setminus(\mathcal{V}_\Gamma\cup\mathcal{V}_{ext})} \alpha_{K,s}\right) \int_K \phi_m(\mathbf{x})d\mathbf{x}, \ K\in\mathcal{M} \\ \phi_\sigma &= \left(1 - \sum_{\mathbf{s}\in\mathcal{V}_\sigma\setminus\mathcal{V}_{ext}} \alpha_{\sigma,\mathbf{s}}\right) d_{f,\sigma} \int_\sigma \phi_f(\mathbf{x})d\tau(\mathbf{x}), \ \sigma\in\mathcal{F}_\Gamma, \\ \phi_\mathbf{s} &= \sum_{\sigma\in\mathcal{F}_{\Gamma,\mathbf{s}}} \alpha_{\sigma,\mathbf{s}} d_{f,\sigma} \int_\sigma \phi_f(\mathbf{x})d\tau(\mathbf{x}), \ \mathbf{s}\in\mathcal{V}_\Gamma\setminus\mathcal{V}_{ext}, \\ \phi_\mathbf{s} &= \sum_{K\in\mathcal{M}_\mathbf{s}} \alpha_{K,\mathbf{s}} \int_K \phi_m(\mathbf{x})d\mathbf{x}, \ \mathbf{s}\in\mathcal{V}\setminus(\mathcal{V}_{ext}\cup\mathcal{V}_\Gamma), \end{aligned}$$

with $d_{f,\sigma} = \frac{\int_\sigma d_f(\mathbf{x})d\tau(\mathbf{x})}{\int_\sigma d\tau(\mathbf{x})}$.

3.2.2 Time integration

For $N\in\mathbb{N}^*$, let us consider the time discretization $t^0 = 0 < t^1 < \cdots < t^{n-1} < t^n \cdots < t^N = T$ of the time interval

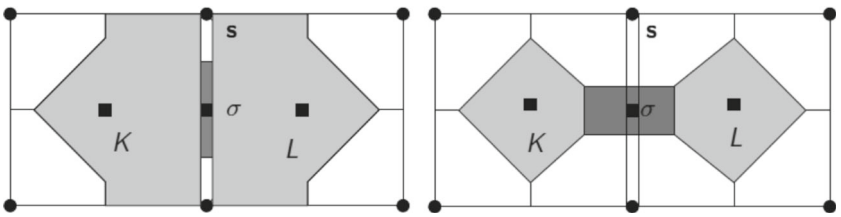


Fig. 4 Example of choices of the control volumes at cells, fracture face, and nodes, in the case of two cells K and L splitted by one fracture face σ (the width of the fracture has been enlarged in this figure). The *left figure* exhibits the good choice with no mixing of fracture and matrix rocktypes while the *right figure* exhibits the bad choice enlarging artificially the fracture

$[0, T]$. We denote the time steps by $\Delta t^n = t^{n+1} - t^n$ for all $n = 0, \cdots, N-1$.

Given $\bar{c}_{\mathbf{s}}, \mathbf{s} \in \mathcal{V}_{ext}$ with arbitrary values on the set of ouput boundary nodes

$$\mathcal{V}_{ext}^{+} = \{\mathbf{s} \in \mathcal{V}_{ext} \mid F_{K,\mathbf{s}}(u_{\mathcal{D}}) \geq 0 \,\forall K \in \mathcal{M}_{\mathbf{s}} \text{ and } F_{\sigma,\mathbf{s}}(u_{\mathcal{D}}) \geq 0 \,\forall \sigma \in \mathcal{F}_{\Gamma,\mathbf{s}}\},$$

and $c_{\mathcal{D}}^0 \in X_{\mathcal{D}}$ such that $c_{\mathbf{s}}^0 = \bar{c}_{\mathbf{s}}$ for all $\mathbf{s} \in \mathcal{V}_{ext}$, the transport discrete model amounts to find $c_{\mathcal{D}}^{n+1} \in X_{\mathcal{D}}$ for all $n = 0, \cdots, N-1$ satisfying the following discrete conservation laws and Dirichlet input conditions

$$\begin{cases} \phi_K \dfrac{c_K^{n+1} - c_K^n}{\Delta t^n} + \displaystyle\sum_{\mathbf{s} \in \mathcal{V}_K} H_{K,\mathbf{s}}(c_{\mathcal{D}}^n) + \sum_{\sigma \in \mathcal{F}_{\Gamma,K}} H_{K,\sigma}(c_{\mathcal{D}}^n) = 0, \ K \in \mathcal{M}, \\ \phi_\sigma \dfrac{c_\sigma^{n+1} - c_\sigma^n}{\Delta t^n} + \displaystyle\sum_{\mathbf{s} \in \mathcal{V}_\sigma} H_{\sigma,\mathbf{s}}(c_{\mathcal{D}}^n) - \sum_{K \in \mathcal{M}_\sigma} H_{K,\sigma}(c_{\mathcal{D}}^n) = 0, \ \sigma \in \mathcal{F}_\Gamma, \\ \phi_{\mathbf{s}} \dfrac{c_{\mathbf{s}}^{n+1} - c_{\mathbf{s}}^n}{\Delta t^n} - \displaystyle\sum_{K \in \mathcal{M}_{\mathbf{s}}} H_{K,\mathbf{s}}(c_{\mathcal{D}}^n) - \sum_{\sigma \in \mathcal{F}_{\Gamma,\mathbf{s}}} H_{\sigma,\mathbf{s}}(c_{\mathcal{D}}^n) = 0, \ \mathbf{s} \in \mathcal{V}_{int}, \\ c_{\mathbf{s}}^{n+1} = \bar{c}_{\mathbf{s}}, \ \mathbf{s} \in \mathcal{V}_{ext}, \end{cases}$$

with the following explicit upwind two point fluxes

$$\begin{aligned} H_{K,\nu}(c_{\mathcal{D}}^n) &= c_K^n F_{K,\nu}(u_{\mathcal{D}})^+ + c_\nu^n F_{K,\nu}(u_{\mathcal{D}})^- \\ H_{\sigma,\mathbf{s}}(c_{\mathcal{D}}^n) &= c_\sigma^n F_{\sigma,\mathbf{s}}(u_{\mathcal{D}})^+ + c_{\mathbf{s}}^n F_{\sigma,\mathbf{s}}(u_{\mathcal{D}})^-. \end{aligned} \tag{8}$$

The solution of this explicit upwind scheme classically satisfies the following maximum principle

$$m \leq c_\mu^{n+1} \leq M \text{ for all } \mu \in \mathcal{V} \cup \mathcal{F}_\Gamma \cup \mathcal{M} \setminus \mathcal{V}_{ext}^{+},$$

with

$$M = \max_{\mu \in \mathcal{V} \cup \mathcal{F}_\Gamma \cup \mathcal{M} \setminus \mathcal{V}_{ext}^{+}} c_\mu^0 \text{ and } m = \min_{\mu \in \mathcal{V} \cup \mathcal{F}_\Gamma \cup \mathcal{M} \setminus \mathcal{V}_{ext}^{+}} c_\mu^0,$$

provided that the following Courant-Friedrichs-Lewy (CFL) condition

$$\Delta t^n \leq \min(\Delta t_{\mathcal{M}}, \Delta t_{\mathcal{F}_\Gamma}, \Delta t_{\mathcal{V}}), \tag{9}$$

is satisfied with

$$\begin{cases} \Delta t_{\mathcal{M}} = \min\limits_{K \in \mathcal{M}} \dfrac{\phi_K}{\sum_{\mathbf{s} \in \mathcal{V}_K} F_{K,\mathbf{s}}(u_{\mathcal{D}})^+ + \sum_{\sigma \in \mathcal{F}_{\Gamma,K}} F_{K,\sigma}(u_{\mathcal{D}})^+}, \\ \Delta t_{\mathcal{F}_\Gamma} = \min\limits_{\sigma \in \mathcal{F}_\Gamma} \dfrac{\phi_\sigma}{\sum_{\mathbf{s} \in \mathcal{V}_\sigma} F_{\sigma,\mathbf{s}}(u_{\mathcal{D}})^+ + \sum_{K \in \mathcal{M}_\sigma} (-F_{K,\sigma}(u_{\mathcal{D}}))^+}, \\ \Delta t_{\mathcal{V}} = \min\limits_{\mathbf{s} \in \mathcal{V}_{int}} \dfrac{\phi_{\mathbf{s}}}{\sum_{K \in \mathcal{M}_{\mathbf{s}}} (-F_{K,\mathbf{s}}(u_{\mathcal{D}}))^+ + \sum_{\sigma \in \mathcal{F}_{\Gamma,\mathbf{s}}} (-F_{\sigma,\mathbf{s}}(u_{\mathcal{D}}))^+}. \end{cases}$$

4 Parallel implementation in ComPASS

The hybrid dimensional Darcy flow and transport discrete model is implemented in the framework of the code computing parallel architecture to speed up simulations (ComPASS) [15], which focuses on parallel high performance simulation (distributed memory, message parsing interface—MPI) adapted to general unstructured polyhedral meshes (see [21]).

4.1 Mesh non overlapping and overlapping decompositions

Let us denote by N_p the number of MPI processes. The set of cells $\mathcal{M}$ is partitioned into N_p subsets $\mathcal{M}^p$, $p = 1, ..., N_p$ using the library METIS [32]. The partitioning of the set of nodes $\mathcal{V}$ and of the set of fracture faces $\mathcal{F}_\Gamma$ is defined as follows: assuming we have defined a global index of the cells $K \in \mathcal{M}$ let us denote by $K(\mathbf{s}), \mathbf{s} \in \mathcal{V}$ (resp. $K(\sigma)$, $\sigma \in \mathcal{F}_\Gamma$) the cell with the smallest global index among those of $\mathcal{M}_{\mathbf{s}}$ (resp. $\mathcal{M}_\sigma$). Then we set

$$\mathcal{V}^p = \{\mathbf{s} \in \mathcal{V} \mid K(\mathbf{s}) \in \mathcal{M}^p\},$$

and

$$\mathcal{F}_\Gamma^p = \{\sigma \in \mathcal{F}_\Gamma \mid K(\sigma) \in \mathcal{M}^p\}.$$

The overlapping decomposition of $\mathcal{M}$ into the sets

$$\overline{\mathcal{M}}^p, \ p = 1, ..., N_p,$$

is chosen in such a way that any compact finite volume scheme such as the VAG scheme can be assembled locally on each process. Hence, as exhibited in Fig. 5, $\overline{\mathcal{M}}^p$ is defined as the set of cells sharing a node with a cell of $\mathcal{M}^p$. The overlapping decompositions of the set of nodes and of the set of fracture faces follow from this definition:

$$\overline{\mathcal{V}}^p = \bigcup_{K \in \overline{\mathcal{M}}^p} \mathcal{V}_K, \ p = 1, \cdots, N_p,$$

and

$$\overline{\mathcal{F}}_\Gamma^p = \bigcup_{K \in \overline{\mathcal{M}}^p} \mathcal{F}_K \cap \mathcal{F}_\Gamma, \ p = 1, \cdots, N_p.$$

The partitioning of the mesh is performed by the master process (process 1), and then, each local mesh is distributed to its process. Therefore, each MPI process contains the local mesh $(\overline{\mathcal{M}}^p, \overline{\mathcal{V}}^p, \overline{\mathcal{F}}_\Gamma^p)$, $p = 1, 2, ..., N_p$ which is splitted into two parts:

own mesh:$(\mathcal{M}^p, \mathcal{V}^p, \mathcal{F}_\Gamma^p)$,

ghost mesh:$(\overline{\mathcal{M}}^p \setminus \mathcal{M}^p, \overline{\mathcal{V}}^p \setminus \mathcal{V}^p, \overline{\mathcal{F}}_\Gamma^p \setminus \mathcal{F}_\Gamma^p)$.

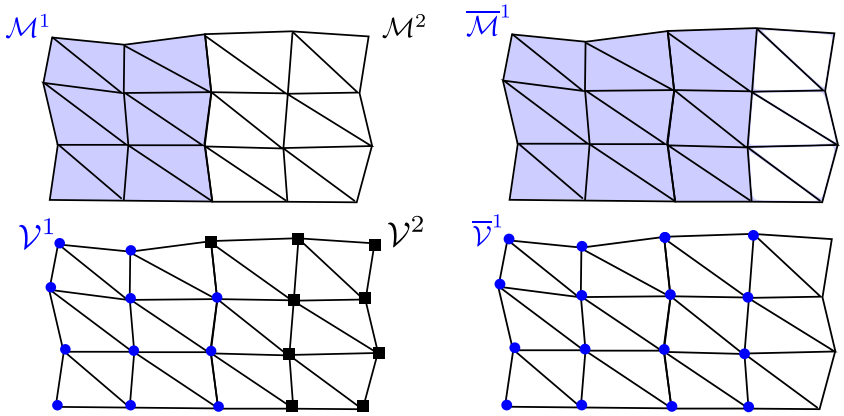


Fig. 5 Example of mesh decomposition

We now turn to the parallel implementation of the discrete hybrid dimensional Darcy flow model (2) and transport model (4).

4.2 Parallelization of the discrete hybrid dimensional Darcy flow

On each process $p = 1, ..., N_p$, the local problem of the discrete hybrid dimensional Darcy flow (2) is defined by the set of unknowns u_μ, $\mu \in \overline{\mathcal{V}}^p \cup \overline{\mathcal{F}}_\Gamma^p \cup \overline{\mathcal{M}}^p$ and the set of equations

$$\begin{cases} \sum_{\mathbf{s}\in\mathcal{V}_K} F_{K,\mathbf{s}}(u_\mathcal{D}) + \sum_{\sigma\in\mathcal{F}_{\Gamma,K}} F_{K,\sigma}(u_\mathcal{D}) = 0, \quad K \in \overline{\mathcal{M}}^p, \\ \sum_{\mathbf{s}\in\mathcal{V}_\sigma} F_{\sigma,\mathbf{s}}(u_\mathcal{D}) + \sum_{K\in\mathcal{M}_\sigma} -F_{K,\sigma}(u_\mathcal{D}) = 0, \quad \sigma \in \mathcal{F}_\Gamma^p, \\ \sum_{K\in\mathcal{M}_s} -F_{K,\mathbf{s}}(u_\mathcal{D}) + \sum_{\sigma\in\mathcal{F}_{\Gamma,\mathbf{s}}} -F_{\sigma,\mathbf{s}}(u_\mathcal{D}) = 0, \quad \mathbf{s} \in \mathcal{V}_{int} \cap \mathcal{V}^p, \\ u_\mathbf{s} = \bar{u}_\mathbf{s}, \quad \mathbf{s} \in \mathcal{V}_{ext} \cap \mathcal{V}^p. \end{cases} \tag{10}$$

Note that this includes the equations of the nodes $\mathbf{s} \in \mathcal{V}^p$, of the fracture faces $\sigma \in \mathcal{F}_\Gamma^p$ and of the cells $K \in \overline{\mathcal{M}}^p$, both those own cells in $K \in \mathcal{M}^p$ and the ghost cells $K \in \overline{\mathcal{M}}^p \setminus \mathcal{M}^p$. The set of equations can be rewritten as the following rectangular linear system

$$\begin{pmatrix} A_{vv}^p & A_{vf}^p & A_{vc}^p \\ A_{fv}^p & A_{ff}^p & A_{fc}^p \\ A_{cv}^p & A_{cf}^p & A_{cc}^p \end{pmatrix} \begin{pmatrix} \overline{U_v^p} \\ \overline{U_f^p} \\ \overline{U_c^p} \end{pmatrix} = \begin{pmatrix} b_v^p \\ b_f^p \\ \overline{b_c^p} \end{pmatrix} \tag{11}$$

where $\overline{U_v^p} \in \mathbb{R}^{\overline{\mathcal{V}}^p}$, $\overline{U_f^p} \in \mathbb{R}^{\overline{\mathcal{F}}_\Gamma^p}$ and $\overline{U_c^p} \in \mathbb{R}^{\overline{\mathcal{M}}^p}$ denote the vector of process p own and ghost unknowns at nodes, fracture faces and cells, respectively. The above matrices have the following sizes

$$\begin{aligned} &A_{vv}^p \in \mathbb{R}^{\mathcal{V}^p \times \overline{\mathcal{V}}^p}, \ A_{vf}^p \in \mathbb{R}^{\mathcal{V}^p \times \overline{\mathcal{F}}_\Gamma^p}, \ A_{vc}^p \in \mathbb{R}^{\mathcal{V}^p \times \overline{\mathcal{M}}^p}, \\ &A_{fv}^p \in \mathbb{R}^{\mathcal{F}_\Gamma^p \times \overline{\mathcal{V}}^p}, \ A_{ff}^p \in \mathbb{R}^{\mathcal{F}_\Gamma^p \times \overline{\mathcal{F}}_\Gamma^p}, \ A_{fc}^p \in \mathbb{R}^{\mathcal{F}_\Gamma^p \times \overline{\mathcal{M}}^p}, \\ &A_{cv}^p \in \mathbb{R}^{\overline{\mathcal{M}}^p \times \overline{\mathcal{V}}^p}, \ A_{cf}^p \in \mathbb{R}^{\overline{\mathcal{M}}^p \times \overline{\mathcal{F}}_\Gamma^p}, \ A_{cc}^p \in \mathbb{R}^{\overline{\mathcal{M}}^p \times \overline{\mathcal{M}}^p}. \end{aligned}$$

and $b_v^p \in \mathbb{R}^{\mathcal{V}^p}$, $b_f^p \in \mathbb{R}^{\mathcal{F}_\Gamma^p}$, $\overline{b_c^p} \in \mathbb{R}^{\overline{\mathcal{M}}^p}$ denote the corresponding right hand side vectors. The matrix A_{cc}^p is a non singular diagonal matrix and the cell unknowns can be easily eliminated without fill-in leading to the following Schur complement system

$$A^p \begin{pmatrix} \overline{U_v^p} \\ \overline{U_f^p} \end{pmatrix} = b^p \tag{12}$$

with

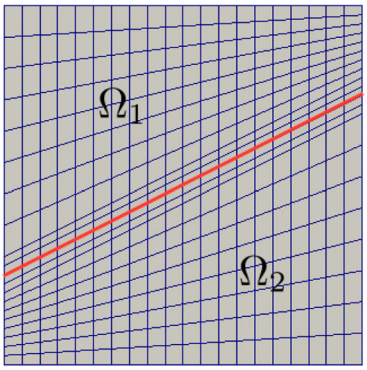
$$\begin{aligned} A^p &:= \begin{pmatrix} A_{vv}^p & A_{vf}^p \\ A_{fv}^p & A_{ff}^p \end{pmatrix} - \begin{pmatrix} A_{vc}^p \\ A_{fc}^p \end{pmatrix} (A_{cc}^p)^{-1} \begin{pmatrix} A_{cv}^p & A_{cf}^p \end{pmatrix}, \\ b^p &:= \begin{pmatrix} b_v^p \\ b_f^p \end{pmatrix} - \begin{pmatrix} A_{vc}^p \\ A_{fc}^p \end{pmatrix} (A_{cc}^p)^{-1} \overline{b_c^p}, \end{aligned}$$

and

$$\overline{U_c^p} = (A_{cc}^p)^{-1} (b_c^p - A_{cv}^p \overline{U_v^p} - A_{cf}^p \overline{U_f^p}). \tag{13}$$

The linear system (12) is built locally on each process p and transfered to the parallel linear solver library PETSc [7] or Trilinos [27]. The parallel matrix and the parallel vector in PETSc or Trilinos are stored in a distributed manner, i.e., each process stores its own rows. We construct the following parallel linear system

$$AU = b, \tag{14}$$



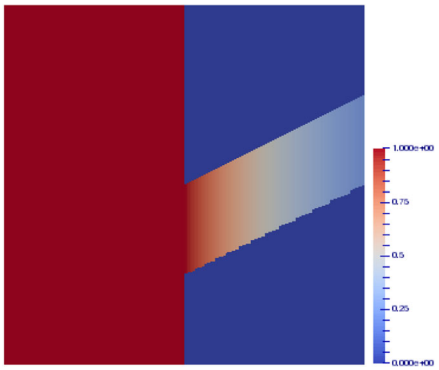


Fig. 6 *Left*: example of mesh with $n_x = 20$ where the *red line* is the fracture. *Right*: analytical solution of (17) at time $t_f = 0.5$

with

$$A := \begin{pmatrix} A^1 R^1 \\ A^2 R^2 \\ \vdots \\ A^{N_p} R^{N_p} \end{pmatrix} \begin{matrix} \} \text{ process } 1 \\ \} \text{ process } 2 \\ \vdots \\ \} \text{ process} N_p \end{matrix},$$

$$U := \begin{pmatrix} U_v^1 \\ U_f^1 \\ U_v^2 \\ U_f^2 \\ \vdots \end{pmatrix} \begin{matrix} \} \text{ process } 1 \\ \} \text{ process } 2 \\ \vdots \end{matrix}, \quad b := \begin{pmatrix} b^1 \\ b^2 \\ \vdots \\ b^{N^p} \end{pmatrix} \begin{matrix} \} \text{ process } 1 \\ \} \text{ process } 2 \\ \vdots \\ \} \text{ process} N_p \end{matrix}$$

where R^p, $p = 1, 2, ..., N_p$ is a restriction matrix satisfying

$$R^p U = \begin{pmatrix} \overline{U_v^p} \\ \overline{U_f^p} \end{pmatrix}.$$

The matrix $A^p R^p$, the vector $\begin{pmatrix} U_v^p \\ U_f^p \end{pmatrix}$ and the vector b^p are stored in process p.

The parallel linear system (14) is solved using the GMRES or BiCGStab algorithm preconditioned by different type of preconditioners as discussed in the numerical section. The solution of the linear system provides on each process p the solution vector $\begin{pmatrix} U_v^p \\ U_f^p \end{pmatrix}$ of own node and fracture-face unknowns. Then, the ghost node unknowns u_μ, $\mu \in (\overline{\mathcal{V}}^p \backslash \mathcal{V}^p)$ and the ghost fracture-face unknowns u_μ, $\mu \in (\overline{\mathcal{F}}_\Gamma^p \backslash \mathcal{F}_\Gamma^p)$ are recovered by a synchronization step with MPI communications. This synchronization is efficiently implemented using a PETSc or Trilinos matrix vector product

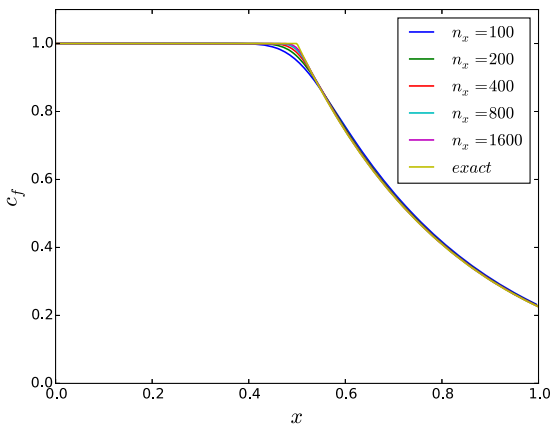
$$\overline{U} = SU \tag{15}$$

where

$$\overline{U} := \begin{pmatrix} \overline{U_v^1} \\ \overline{U_f^1} \\ \overline{U_v^2} \\ \overline{U_f^2} \\ \vdots \end{pmatrix}$$

is the vector of own and ghost node and fracture-face unknowns on all processes. The matrix S, containing only 0 and 1 entries, is assembled once and for all at the beginning of the simulation.

Finally, thanks to Eq. 13, the vector of own and ghost cell unknowns $\overline{U_c^p}$ is computed locally on each process p.



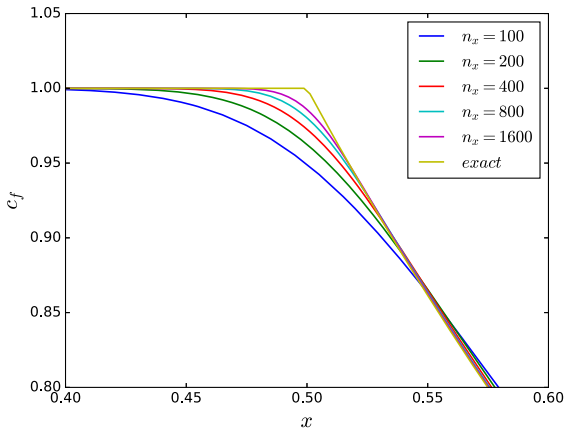


Fig. 7 *Left*: analytical solution and numerical solutions along the fracture at time t_f with $n_x = 100, 200, 400, 800, 1600$. *Right*: zoom view of left figure around $x = 0.5$

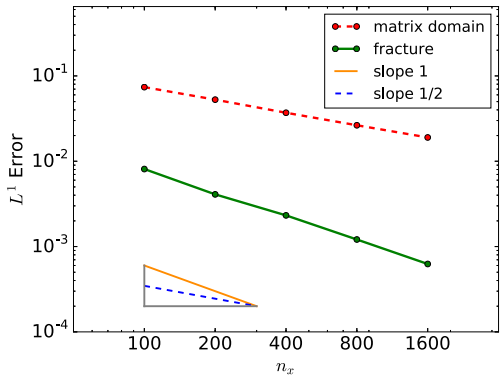


Fig. 8 Relative L^1 errors and references in the matrix domain and in the fracture at time t_f between the analytical solution and the numerical solutions as a function of the grid size $n_x = 100, 200, 400, 800, 1600$

In conclusion, the parallel implementation of the discrete hybrid dimensional Darcy flow can be summarized as:

Algorithm 1 Parallel implementation of the discrete hybrid dimensional Darcy flow

1: Assemble locally on each process the rectangular linear system (11).
2: Compute locally on each process the Schur complement (12) of (11).
3: Construct the parallel linear system (14) in PETSc or Trilinos.
4: Solve the parallel linear system (14) to obtain the solution at own nodes and fracture faces.
5: Communicate the solution at ghost nodes and fracture faces from (15).
6: Compute locally on each process the solution at own and ghost cells from (13).

4.3 Parallelization of the discrete hybrid dimensional transport model

The parallel implementation of the transport model (4) with an explicit upwind discretization of the fluxes consists of the following four steps.

1. Compute the Darcy matrix and fracture fluxes defined by Eqs. 6 and 7.
2. Compute the maximum time step Δt satisfying the CFL condition (9) and set $\Delta t^n = \Delta t$ for all $n = 0, \cdots, N-2$, and $\Delta t^{N-1} = T - (N-1)\Delta t$ with $N = \lceil \frac{T}{\Delta t} \rceil$.
3. For each time step $n = 0, 1, ..., N-2$,

(a) Compute $c_{\mathbf{s}}^{n+1}$, c_{σ}^{n+1} and c_K^{n+1}, $\mathbf{s} \in \mathcal{V}_{int} \cap \mathcal{V}^p$, $\sigma \in \mathcal{F}_\Gamma^p$, $K \in \overline{\mathcal{M}}^p$ solution of the following explicit equations

$$\begin{cases} \phi_K \dfrac{c_K^{n+1} - c_K^n}{\Delta t} + \displaystyle\sum_{\mathbf{s} \in \mathcal{V}_K} H_{K,\mathbf{s}}(c_{\mathcal{D}}^n) + \sum_{\sigma \in \mathcal{F}_{\Gamma,K}} H_{K,\sigma}(c_{\mathcal{D}}^n) = 0, \quad K \in \overline{\mathcal{M}}^p, \\ \phi_\sigma \dfrac{c_\sigma^{n+1} - c_\sigma^n}{\Delta t} + \displaystyle\sum_{\mathbf{s} \in \mathcal{V}_\sigma} H_{\sigma,\mathbf{s}}(c_{\mathcal{D}}^n) - \sum_{K \in \mathcal{M}_\sigma} H_{K,\sigma}(c_{\mathcal{D}}^n) = 0, \quad \sigma \in \mathcal{F}_\Gamma^p, \\ \phi_{\mathbf{s}} \dfrac{c_{\mathbf{s}}^{n+1} - c_{\mathbf{s}}^n}{\Delta t} - \displaystyle\sum_{K \in \mathcal{M}_{\mathbf{s}}} H_{K,\mathbf{s}}(c_{\mathcal{D}}^n) - \sum_{\sigma \in \mathcal{F}_{\Gamma,\mathbf{s}}} H_{\sigma,\mathbf{s}}(c_{\mathcal{D}}^n) = 0, \quad \mathbf{s} \in \mathcal{V}_{int} \cap \mathcal{V}^p, \\ c_{\mathbf{s}} = \bar{c}_{\mathbf{s}}, \quad \mathbf{s} \in \mathcal{V}_{ext} \cap \mathcal{V}^p. \end{cases} \tag{16}$$

(b) Get the node and fracture-face ghost unknowns $c_{\mathbf{s}}^{n+1}$, c_σ^{n+1}, $\mathbf{s} \in \mathcal{V}_{int} \cap (\overline{\mathcal{V}}^p \backslash \mathcal{V}^p)$, $\sigma \in \overline{\mathcal{F}}_\Gamma^p \backslash \mathcal{F}_\Gamma^p$ using the PETSc or Trilinos matrix vector product with the matrix S defined in Eq. 15, as was done for the Darcy flow solution U:

$$\begin{pmatrix} \overline{C_v^1} \\ \overline{C_f^1} \\ \overline{C_v^2} \\ \overline{C_f^2} \\ \vdots \end{pmatrix} = S \begin{pmatrix} C_v^1 \\ C_f^1 \\ C_v^2 \\ C_f^2 \\ \vdots \end{pmatrix}$$

Thanks to our mesh decomposition, step 1 and step 3a are performed locally on each process. For step 2, the maximum time step Δt^p is computed locally on each process p, then the time step Δt is obtained using the MPI reduce operation.

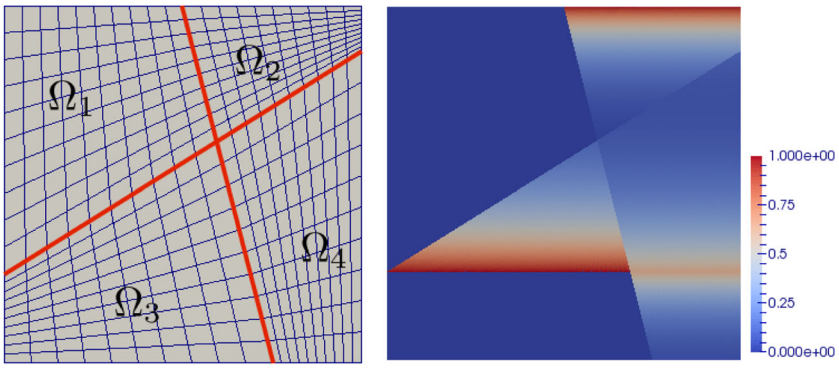


Fig. 9 *Left*: example of mesh with $n_x = 20$ where the red lines account for the four fractures. *Right*: stationary analytical solution of Eq. 18

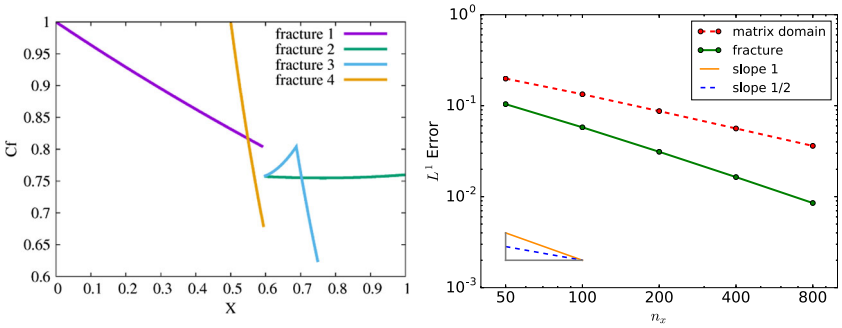


Fig. 10 *Left*: stationary analytical solution in the four fractures as a function of the x coordinate. *Right*: relative L^1 errors and references in the matrix domain and in the fracture network between the stationary analytical and numerical solutions as a function of the grid size $n_x = 100, 200, 400, 800$

5 Numerical experiments

All the numerical tests have been implemented on the Cicada cluster of the University Nice Sophia Antipolis consisting of 72 nodes (16 cores/node, Intel Sandy Bridge E5-2670, 64GB/node). We always fix 1core per process and 16 processes per node. Thecommunications are handled by OpenMPI 1.8.2 (GCC 4.9) andPETSc 3.5.3.

The first two test casesare designed in order to validate the Darcy fluxes and theconvergence of the transport model discretization on twoanalytical solutions including one fracture for the first testcase and four intersecting fractures for the secondtest case. In the remaining test cases, the parallel scalabilityof our Darcy flow and transport solvers is assessedwith different types of fracture networks and meshes anddifferent matrix fracture permeability ratios. In particular,the last test case applies our algorithm to a complexfracture network with hundreds of fractures.

5.1 Numerical convergence for an analytical solution with one fracture

Let us set $\Omega = (0,1)^2$, and denote by (x, y) the Cartesian coordinates of $\mathbf{x}$. We then define $\mathbf{x}_1 = \left(0, \frac{1}{4}\right)$, $\theta \in \left(0, \arctan\left(\frac{3}{4}\right)\right)$, $\mathbf{x}_2 = \left(1, \frac{1}{4} + \tan(\theta)\right)$. Let $\Omega_1 = \left\{(x, y) \in \Omega \mid y > \frac{1}{4} + x\tan(\theta)\right\}$, and $\Omega_2 = \Omega \setminus \overline{\Omega}_1$.We consider a single fracture defined by $\Gamma = (\mathbf{x}_1, \mathbf{x}_2) = \partial\Omega_1 \cap \partial\Omega_2$ with tangential permeability$\Lambda_f > 0$, and width $d_f > 0$. The matrix permeability is isotropic and set to $\Lambda_m = 1$, the matrix and fracture porosities are setto $\phi_m = \phi_f = 1$, and the fluid viscosity is set to $\mu = 1$. The pressure solution isfixed to $u(x, y) = 1 - x$. In this case, the transport model (4) reduces to the following system of equations which specifies our choice of the boundary and initial conditions:

$$
\begin{cases}
\partial_t c_{m,\alpha}(x,y,t) + \partial_x c_{m,\alpha}(x,y,t) = 0 & \text{on } \Omega_\alpha \times (0,T),\ \alpha = 1,2,\\
c_{m,\alpha}(x,y,0) = 0 & \text{on } \Omega_\alpha,\ \alpha = 1,2,\\
c_{m,1}(0,y,t) = 1 & \text{on } \left(\frac{1}{4},1\right)\times(0,T),\\
c_{m,2}(0,y,t) = 1 & \text{on } \left(0,\frac{1}{4}\right)\times(0,T),\\
c_{m,2}(x,\frac{1}{4}+x\tan(\theta),t) = c_f(x,t) & \text{on } (0,1)\times(0,T),\\
\mathcal{L}c_f(x,t) = \beta c_{m,1}\left(x,\frac{1}{4}+x\tan(\theta),t\right) & \text{on } (0,1)\times(0,T),\\
c_f(0,t) = 1 & \text{on } (0,T),\\
c_f(x,0) = 0 & \text{on } (0,1),
\end{cases}
\tag{17}
$$

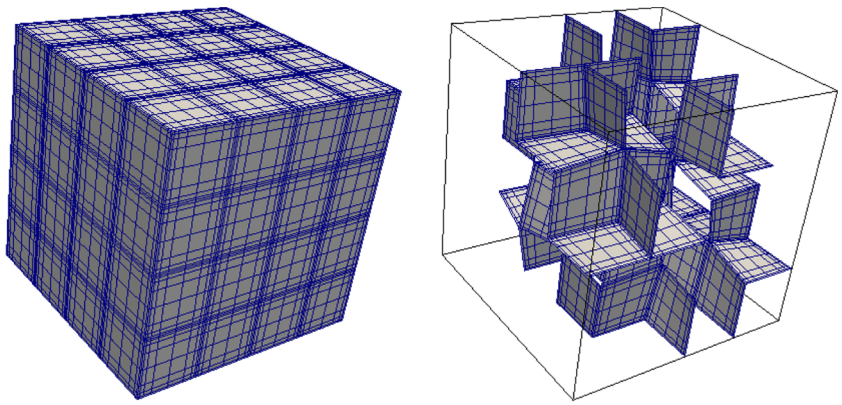


Fig. 11 Hexahedral mesh in the matrix domain (*left*) and in the fracture network (*right*) with $n_x = 32$

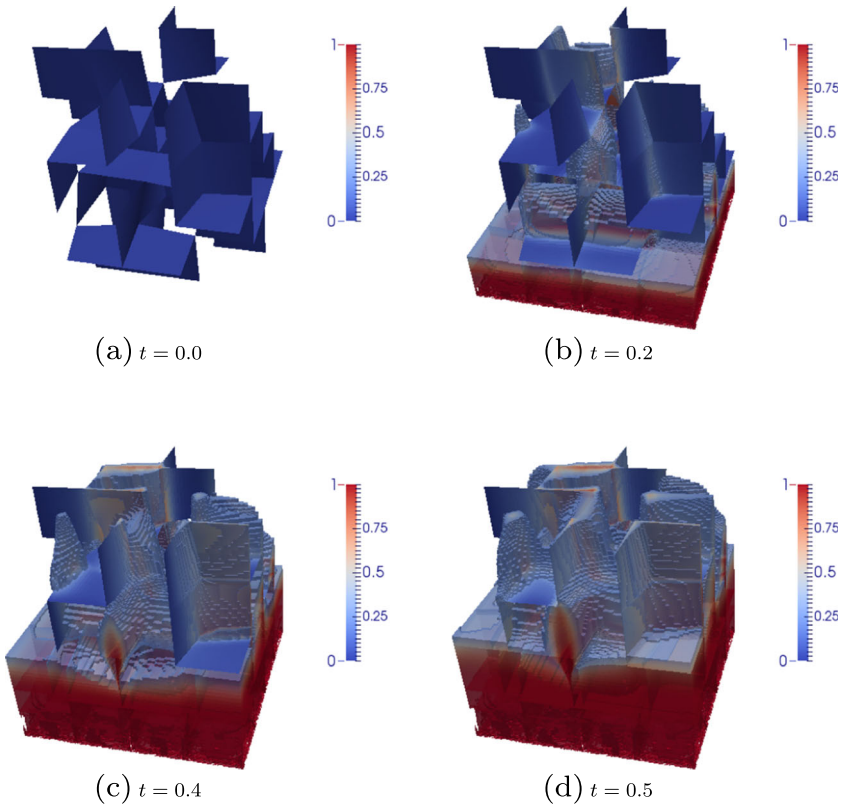


Fig. 12 Concentration in the matrix domain and in the fracture network obtained at different times with the mesh $n_x = 128$. A threshold concentration of 0.2 is used in the matrix domain

where $\mathcal{L} = \partial_t + k\partial_x + \beta$ with $\beta = \frac{\sin(\theta)}{d_f}$ and $k = \Lambda_f \cos^2(\theta)$. It is assumed that $k > 1$. This system can be integrated along the characteristics of the matrix and fracture velocity fields leading to the following analytical solution:

$$c_{m,1}(x, y, t) = \begin{cases} 0 \text{ if } t < x, \\ 1 \text{ if } t > x, \end{cases}$$

$$c_f(x, t) = \begin{cases} 0 \text{ if } t < \frac{x}{k}, \\ e^{-\frac{\beta}{k-1}(x-t)} \text{ if } \frac{x}{k} < t < x, \\ 1 \text{ if } t > x, \end{cases}$$

$$c_{m,2}(x, y, t) = \begin{cases} \text{if} y \in (0, \frac{1}{4}) \begin{cases} 0 \text{ if } t < x, \\ 1 \text{ if } t > x, \end{cases} \\ \text{if} y \in (\frac{1}{4}, \frac{1}{4} + \tan(\theta)) \begin{cases} 0 \text{ if } t < x - \frac{4y-1}{4\tan(\theta)}, \\ c_f\left(\frac{4y-1}{4\tan(\theta)}, t + \frac{4y-1}{4\tan(\theta)} - x\right) \text{ if } t > x - \frac{4y-1}{4\tan(\theta)}. \end{cases} \end{cases}$$

Table 1 Number of linear solver iterations vs. the number of MPI processes obtained with different linear solvers and preconditioners for the mesh size $n_x = 128$

N_p	2	4	8	16	32	64	128	256	512
GMRES + Boomer AMG	15	15	15	15	15	16	15	15	15
GMRES + Aggregation AMG	59	78	65	39	65	54	73	53	62
GMRES + ILU(0)	751	707	655	644	648	634	633	624	613
BiCGStab + Boomer AMG	9	9	9	9	9	10	9	9	10
BiCGStab + ILU(0)	508	476	484	503	473	513	491	487	484

Table 2 Linear solver setup phase and solve phase computation times vs. the number of MPI processes obtained with different linear solvers and preconditioners for the mesh size $n_x = 128$

N_p		2	4	8	16	32	64	128	256	512
GMRES Boomer AMG	Setup	34.1	20.1	16.3	11.7	11.3	11.9	12.1	19.2	29.3
	Solve	26.3	15.6	14.8	7.2	5.2	3.8	2.5	5.2	9.6
GMRES Aggregation AMG	Setup	4.7	1.9	1.6	1.5	2.3	2.9	4.4	6.6	11.3
	Solve	45.1	20.9	17.0	9.7	5.2	2.5	2.3	1.5	3.2
GMRES ILU(0)	Setup	16.9	21.3	16.3	23.2	14.6	11.0	9.7	6.0	4.8
	Solve	672.3	590.9	281.6	163.9	71.4	30.7	16.7	8.3	4.0
BiCGStab Boomer AMG	Setup	38.0	23.3	15.0	10.3	9.1	9.4	12.8	14.8	23.8
	Solve	37.1	21.3	11.5	7.4	4.1	2.9	2.5	4.4	10.0
BiCGStab ILU(0)	Setup	18.9	19.9	16.5	22.1	14.3	12.4	9.4	5.8	3.9
	Solve	179.4	111.7	86.0	59.9	27.9	15.4	8.0	4.2	2.2

In the following numerical experiments, the parameters are set to $\tan(\theta) = \frac{1}{2}$, $\Lambda_f = 20$ and $d_f = 0.01$. The mesh is a topologically Cartesian $n_x \times n_x$ grid. Figure 6 shows an example of the mesh with $n_b = 20$ as well as the analytical solution in the matrix obtained at time $t_f = 0.5$ chosen as the final time of the simulation. The time step is defined by the maximum time step allowed by the CFL condition (9). Figure 8 exhibits the convergence of the relative L^1 errors between the analytical solution and the numerical solution at time t_f both in the matrix domain and in the fracture as a function of the grid size $n_x = 100, 200, 400, 800, 1600$. Figure 7 shows the analytical solution and the numerical solutions obtained at time t_f along the fracture. In both cases, we can observe the expected convergence of the numerical solution to the analytical solution with a higher order of convergence in the fracture due to the fact that at time t_f the analytical solution in the fracture is continuous as exhibited in Fig. 7.

5.2 Numerical convergence for an analytical solution with four intersecting fractures

Let $\Omega = (0, 1)^2$, $\mathbf{x}_1 = \left(0, \frac{1}{4}\right)$, $\theta_1 \in \left(0, \arctan\left(\frac{3}{4}\right)\right)$, $\mathbf{x}_2 = \left(1, \frac{1}{4} + \tan(\theta_1)\right)$, $\mathbf{x}_3 = \left(\frac{3}{4}, 0\right)$, $\mathbf{x}_4 = \left(\frac{3}{4} - \tan(\theta_2), 1\right)$, and the intersection of $\mathbf{x}_1\mathbf{x}_2$ and $\mathbf{x}_3\mathbf{x}_4$ equal to

$$\mathbf{x}_0 = (x_0, y_0) = \frac{1}{4(1 + \tan(\theta_1)\tan(\theta_2))}(3 - \tan(\theta_2), 1 + 3\tan(\theta_1)).$$

It is assumed that $\theta_1, \theta_2 \in \left(0, \arctan\left(\frac{3}{4}\right)\right)$.

We consider the four fractures $\Gamma_1 = (\mathbf{x}_1, \mathbf{x}_0)$, $\Gamma_2 = (\mathbf{x}_0, \mathbf{x}_2)$, $\Gamma_3 = (\mathbf{x}_3, \mathbf{x}_0)$, $\Gamma_4 = (\mathbf{x}_0, \mathbf{x}_4)$, with tangential permeabilities $\Lambda_{f,1} = \Lambda_{f,2} > 0$, and $\Lambda_{f,3} = \Lambda_{f,4} > 0$, and with widths $d_{f,1} = d_{f,2} > 0$, and $d_{f,3} = d_{f,4} > 0$. It is assumed that $\Lambda_m = 1$.

The fractures partition the domain Ω in the following four subdomains

$$\Omega_1 = \left\{\mathbf{x} = (x, y) \in \Omega \,|\, y > \frac{1}{4} + x\tan(\theta_1),\ x < \frac{3}{4} - y\tan(\theta_2)\right\},$$

$$\Omega_2 = \left\{\mathbf{x} = (x, y) \in \Omega \,|\, y > \frac{1}{4} + x\tan(\theta_1),\ x > \frac{3}{4} - y\tan(\theta_2)\right\},$$

$$\Omega_3 = \left\{\mathbf{x} = (x, y) \in \Omega \,|\, y < \frac{1}{4} + x\tan(\theta_1),\ x < \frac{3}{4} - y\tan(\theta_2)\right\},$$

$$\Omega_4 = \left\{\mathbf{x} = (x, y) \in \Omega \,|\, y < \frac{1}{4} + x\tan(\theta_1),\ x > \frac{3}{4} - y\tan(\theta_2)\right\}.$$

Let us set $\beta_1 = \frac{\sin(\theta_1)}{d_{f,1}}$, $k_1 = \Lambda_{f,1}\cos^2(\theta_1)$, $\beta_2 = \frac{\cos(\theta_2)}{d_{f,3}}$, $k_2 = \Lambda_{f,3}\cos(\theta_2)\sin(\theta_2)$, $r = \frac{\Lambda_{f,3}d_{f,3}\sin(\theta_2)}{\Lambda_{f,1}d_{f,1}\cos(\theta_1)}$. It is assumed that $k_1 > 1$ and $k_2\tan(\theta_2) > 1$. The matrix and fracture porosities are set to $\phi_m = \phi_f = 1$ and the fluid viscosity is set to $\mu = 1$ (Fig. 8).

Table 3 Number of linear solver iterations vs. the matrix fracture permeability ratio $\frac{\Lambda_f}{\Lambda_m}$ for $n_x = 128$ and $N_p = 2, 128$

	$N_p = 2$			$N_p = 128$		
Λ_f/Λ_m	20	100	1000	20	100	1000
GMRES + Boomer AMG	15	15	16	15	15	15
GMRES + Aggregation AMG	59	–	–	73	–	–
GMRES + ILU(0)	751	–	–	633	–	–

-The solver does not converge in 1200 iterations

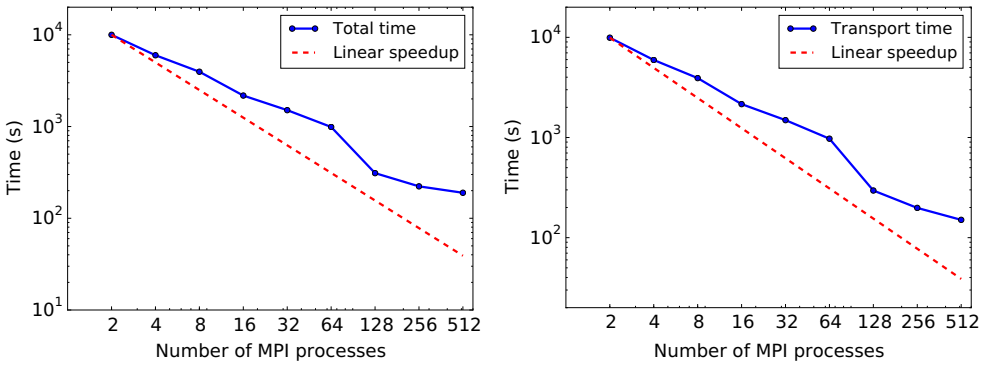


Fig. 13 Total computation time (*left*) and computation time for the transport model (*right*) vs. the number of MPI processes for the mesh size $n_x = 128$

The pressure solution is set to $u(x, y) = 1 - x$. In that case, the transport model (4) reduces to the following system of equations which specifies our choice of the boundary and initial conditions: find $c_{m,\alpha}(x, y, t)$, $\alpha = 1, \cdots, 4$, $c_{f,1}(x, t)$, $c_{f,2}(x, t)$, $c_{f,3}(y, t)$, $c_{f,4}(y, t)$, and $c_0(t)$ such that

$$
\begin{cases}
\partial_t c_{m,\alpha}(x, y, t) + \partial_x c_{m,\alpha}(x, y, t) = 0 & \text{on } \Omega_\alpha \times (0, T),\ \alpha = 1, \cdots, 4, \\
c_{m,\alpha}(x, y, 0) = 0 & \text{on } \Omega_\alpha,\ \alpha = 1, \cdots, 4, \\
c_{m,1}(0, y, t) = 0 & \text{on } \left(\tfrac{1}{4}, 1\right) \times (0, T), \\
c_{m,3}(0, y, t) = 0 & \text{on } \left(0, \tfrac{1}{4}\right) \times (0, T), \\
c_{m,2}\left(\tfrac{3}{4} - y\tan(\theta_2), y, t\right) = c_{f,4}(y, t) & \text{on } (y_0, 1) \times (0, T), \\
c_{m,4}\left(\tfrac{3}{4} - y\tan(\theta_2), y, t\right) = c_{f,3}(y, t) & \text{on } (0, y_0) \times (0, T), \\
c_{m,3}\left(x, \tfrac{1}{4} + x\tan(\theta_1), t\right) = c_{f,1}(x, t) & \text{on } (0, x_0) \times (0, T), \\
c_{m,4}\left(x, \tfrac{1}{4} + x\tan(\theta_1), t\right) = c_{f,2}(x, t) & \text{on } (x_0, 1) \times (0, T), \\
\mathcal{L}_1 c_{f,1}(x, t) - \beta_1 c_{m,1}\left(x, \tfrac{1}{4} + x\tan(\theta_1), t\right) = 0 & \text{on } (0, x_0) \times (0, T), \\
\mathcal{L}_1 c_{f,2}(x, t) - \beta_1 c_{m,2}\left(x, \tfrac{1}{4} + x\tan(\theta_1), t\right) = 0 & \text{on } (x_0, 1) \times (0, T), \\
\mathcal{L}_2 c_{f,3}(y, t) - \beta_2 c_{m,3}\left(\tfrac{3}{4} - y\tan(\theta_2), y, t\right) = 0 & \text{on } (0, y_0) \times (0, T), \\
\mathcal{L}_2 c_{f,4}(y, t) - \beta_2 c_{m,1}\left(\tfrac{3}{4} - y\tan(\theta_2), y, t\right) = 0 & \text{on } (y_0, 1) \times (0, T), \\
c_{f,2}(x_0, t) = c_{f,3}(y_0, t) = c_0(t) & \text{on } (0, T), \\
(r + 1)c_0(t) - c_{f,1}(x_0, t) - r c_{f,4}(y_0, t) = 0 & \text{on } (0, T), \\
c_{f,1}(0, t) = c_{f,4}(1, t) = 1 & \text{on } (0, T), \\
c_{f,1}(x, 0) = 0 & \text{on } (0, x_0), \\
c_{f,2}(x, 0) = 0 & \text{on } (x_0, 1), \\
c_{f,3}(y, 0) = 0 & \text{on } (0, y_0), \\
c_{f,4}(y, 0) = 0 & \text{on } (y_0, 1),
\end{cases}
\tag{18}
$$

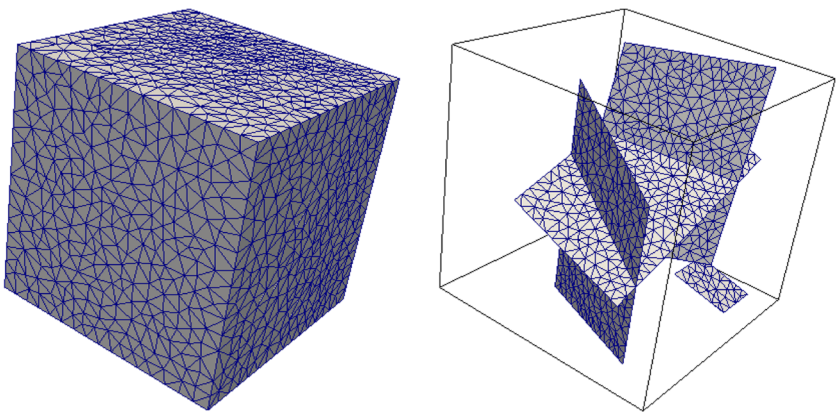


Fig. 14 Example of tetrahedral mesh of the matrix domain (*left*) conforming to the fracture network (*right*)

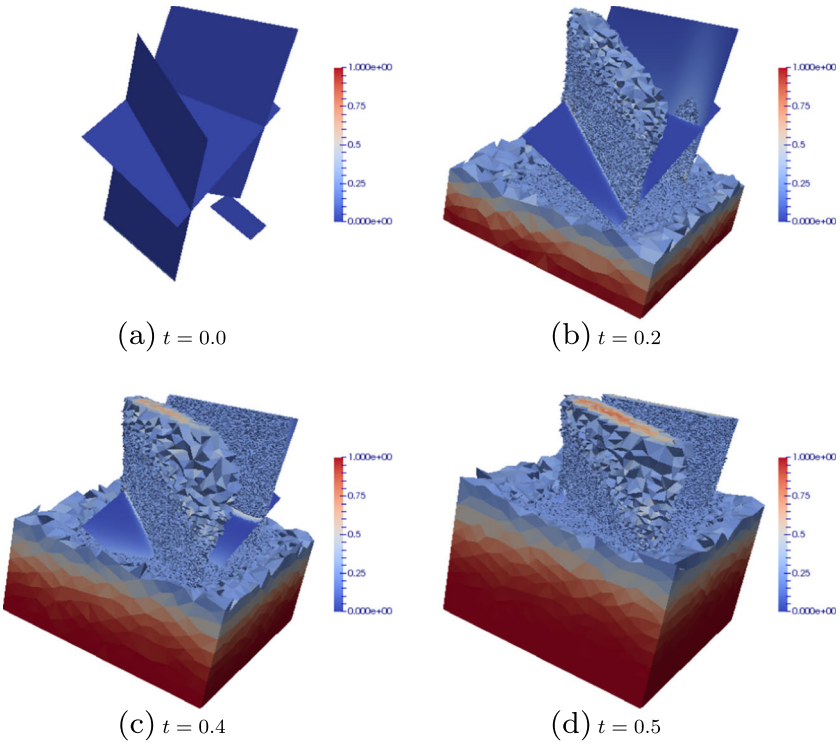


Fig. 15 Concentration in the matrix domain and in the fracture network obtained at different times for the tetrahedral mesh. A threshold of 0.2 is used in the matrix domain

where $\mathcal{L}_1 = \partial_t + k_1\partial_x + \beta_1$ and $\mathcal{L}_2 = \partial_t - k_2\partial_y + \beta_2$.

This system can also be integrated analytically along the characteristics of the matrix and fracture velocity fields, but it leads to complex computations. It is much easier to obtain the stationary solution of this system which is defined in the fractures by

$$
\begin{aligned}
c_{f,1}(x) &= e^{-\frac{\beta_1}{k_1}x},\\
c_{f,4}(y) &= e^{-\frac{\beta_2}{k_2}(1-y)},\\
c_0 &= \frac{e^{-\frac{\beta_1}{k_1}x_0} + re^{-\frac{\beta_2}{k_2}(1-y_0)}}{r+1},\\
c_{f,2}(x) &= e^{-\frac{\beta_1}{k_1}x}\left(c_0 e^{\frac{\beta_1}{k_1}x_0} + \frac{\beta_1}{k_1 r_1}\left(e^{\left(r_1 x - \frac{3\beta_2}{4k_2}\right)} - e^{\left(r_1 x_0 - \frac{3\beta_2}{4k_2}\right)}\right)\right),\\
c_{f,3}(y) &= \begin{cases} e^{\frac{\beta_2}{k_2}y}\left(c_0 e^{-\frac{\beta_2}{k_2}y_0} + \frac{\beta_2}{k_2 r_2}\left(e^{(-\frac{r_2}{4} - \frac{\beta_1}{4k_1\tan(\theta_1)})} - e^{(-r_2 y_0 - \frac{\beta_1}{4k_1\tan(\theta_1)})}\right)\right), & \text{if } y < \frac{1}{4},\\ e^{\frac{\beta_2}{k_2}y}\left(c_0 e^{-\frac{\beta_2}{k_2}y_0} + \frac{\beta_2}{k_2 r_2}\left(e^{(-r_2 y - \frac{\beta_1}{4k_1\tan(\theta_1)})} - e^{(-r_2 y_0 - \frac{\beta_1}{4k_1\tan(\theta_1)})}\right)\right) & \text{if } y > \frac{1}{4}, \end{cases}
\end{aligned}
$$

Table 4 Number of linear solver iterations vs. the number of MPI processes obtained with different linear solvers and preconditioners for the tetrahedral mesh

N_p	2	4	8	16	32	64	128	256	512
GMRES + Boomer AMG	11	12	12	12	12	12	12	12	12
GMRES + Aggregation AMG	38	78	40	39	52	–	35	–	52
GMRES + ILU(0)	1003	725	717	682	667	656	644	629	612
BiCGStab + Boomer AMG	8	7	8	8	8	8	8	8	8
BiCGStab + ILU(0)	565	513	527	544	535	483	489	483	473

-The relative residual norm stagnates after a few iterations
Some future investigations are necessary

Table 5 Linear solver setup phase and solve phase computation times vs. the number of MPI processes obtained with different linear solvers and preconditioners for the tetrahedral mesh

N_p		2	4	8	16	32	64	128	256	512
GMRES Boomer AMG	Setup time	12.4	7.8	4.9	5.0	4.3	6.2	7.2	13.5	22.4
	Solve time	8.0	5.5	2.9	1.7	1.1	0.9	1.4	3.1	6.9
GMRES Aggregation AMG	Setup time	3.7	1.9	1.2	1.8	2.1	1.6	2.9	3.3	4.7
	Solve time	19.7	20.9	5.1	2.7	2.0	-	1.5	-	3.0
GMRES ILU(0)	Setup time	5.7	7.4	7.2	5.6	4.7	5.2	3.4	2.8	1.8
	Solve time	560.6	254.4	150.0	66.5	30.1	15.2	7.7	4.1	2.8
BiCGStab Boomer AMG	Setup time	21.5	14.5	9.9	6.5	5.3	5.9	8.2	12.4	19.7
	Solve time	24.0	10.2	6.1	3.5	1.8	1.5	2.1	4.3	9.5
BiCGStab ILU(0)	Setup time	5.8	6.4	6.4	5.4	4.7	5.0	3.4	2.6	1.8
	Solve time	110.4	63.0	39.0	19.2	11.6	5.4	2.8	1.4	1.2

-The residual norm stagnates after a few iterations

with $r_1 = \frac{\beta_1}{k_1} + \frac{\beta_2}{k_2}\tan(\theta_1)$ and $r_2 = \frac{\beta_2}{k_2} + \frac{\beta_1}{k_1 \tan(\theta_1)}$, and in the matrix by

$$c_{m,1}(x, y) = 0,$$

$$c_{m,2}(x, y) = c_{f,4}(y),$$

$$c_{m,3}(x, y) = \begin{cases} 0 & \text{if } y < \frac{1}{4}, \\ c_{f,1}\left(\frac{y-\frac{1}{4}}{\tan(\theta_1)}\right) & \text{if } y > \frac{1}{4}, \end{cases}$$

$$c_{m,4}(x, y) = \begin{cases} c_{f,3}(y) & \text{if } y < y_0, \\ c_{f,2}\left(\frac{y-\frac{1}{4}}{\tan(\theta_1)}\right) & \text{if } y > y_0. \end{cases}$$

In the following numerical experiments, the parameters are set to $\tan(\theta_1) = \frac{5}{8}$, $\tan(\theta_2) = \frac{1}{4}$, $\Lambda_{f,1} = 200$, $\Lambda_{f,3} = 400$, and $d_{f,1} = d_{f,3} = 0.01$. The mesh is, as for the previous test case, a topologically Cartesian $n_x \times n_x$ grid exhibited in Fig. 9 for $n_x = 20$. Figure 9 also shows the stationary analytical solution in the matrix. The time step is again defined by the maximum time step allowed by the CFL condition (9) and the simulation time is chosen large enough to obtain the numerical stationary solution.

Figure 10 exhibits the convergence of the relative L^1 errors between the stationary analytical and the numerical solutions both in the matrix domain and in the fracture as a function of the grid size $n_x = 100, 200, 400, 800$. We can again observe the expected convergence of the numerical solution to the analytical solution with a higher order of convergence in the fracture network due to the fact that the solution is continuous on each individual fracture as exhibited in Fig. 10. This property is always true when looking at the solution at the matrix time scale and could be exploited in a future work by using an implicit time integration in the fracture coupled to an explicit time integration in the matrix domain with a higher order discretization in space in the spirit of what has been done in [29].

5.3 Fracture network with hexahedral meshes

The objective of this subsection and of the next subsection is to investigate the parallel scalability of the algorithms described in Section 4. In this subsection we consider a topologically Cartesian mesh of size $n_x \times n_x \times n_x$ of the cubic domain $(0, 1)^3$ exhibited in Fig. 11 for $n_x = 32$. The mesh is exponentially refined at the interface between the matrix and the fracture network exhibited in Fig. 12. The permeabilities are isotropic and set to $\Lambda_f = 20$ in the fracture network and to $\Lambda_m = 1$ in the matrix. The porosities are set to $\phi_m = \phi_f = 1$ and the fluid viscosity is set to $\mu = 1$. The initial concentration is set to 0 both in the matrix domain and in the fracture network and a value of 1 of the

Table 6 Number of linear solver iterations vs. the matrix fracture permeability ratio $\frac{\Lambda_f}{\Lambda_m}$ for the tetrahedral mesh and $N_p = 2, 128$

	$N_p = 2$			$N_p = 128$		
Λ_f/Λ_m	20	100	1000	20	100	1000
GMRES + Boomer AMG	11	13	12	12	13	12
GMRES + Aggregation AMG	38	–	–	35	–	–
GMRES + ILU(0)	1002	–	–	644	–	–

-The solver does not converge in 1200 iterations

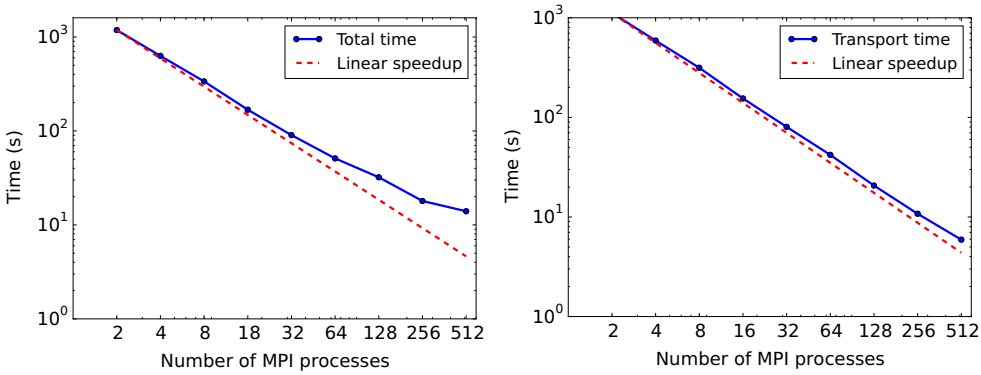


Fig. 16 Total computation time (*left*) and computation time for transport model (*right*) vs. number of MPI processes with tetrahedral mesh

concentration is injected on the bottom boundaries of the matrix and of the fracture network. The pressure is fixed to $u = 1$ and $\gamma u = 1$ on the bottom boundary and to $u = 0$ and $\gamma u = 0$ on the top boundary. The remaining lateral boundaries are considered impervious. Figure 12 exhibits the tracer concentrations obtained with the mesh $n_x = 128$ at times $t = 0$, $t = 0.2$, $t = 0.4$ and at the final simulation time $t_f = 0.5$.

Table 1 presents the numbers of linear solver iterations for the stationary pressure solution for a number of MPI processes ranging from $N_p = 2$ to $N_p = 512$ and with the mesh size $n_x = 128$ corresponding to roughly 2.1×10^6 cells, 2.1×10^6 nodes and 5.2×10^4 fractures faces. Both the GMRES and BiCGStab linear solvers from the PETSc library are tested combined with either the Boomer AMG preconditioner from the Hypre library [30], the Aggregation AMG preconditioner from the Trilinos library [27] or the block Jacobi ILU(0) preconditioner from the Euclid library. No restart is used for the GMRES linear solver. Table 2 shows the corresponding computation times both for the setup phase of the preconditioner and for the solve phase of the linear solver.

According to these tables, the GMRES and the BiCGStab linear solvers combined with the Boomer AMG preconditioner are good choices for a number of processes $N_p \leqslant 128$, while the BiCGStab linear solver combined with the block Jacobi ILU(0) preconditioner is more efficient for this test case for $N_p = 256$ and $N_p = 512$. This was expected since the Boomer AMG preconditioner requires a sufficiently large number of unknowns per core to maintain a good parallel scalability due to the high level of communications in particular in the setup phase of the preconditioner. For this linear system, the number of unknowns per MPI process is roughly 4100 for $N_p = 512$ which is too small for this type of preconditioner while the block Jacobi preconditioner still maintains a good parallel scalability for such a number of unknowns per MPI process. On the other hand, as shown in Table 3, Boomer AMG exhibits an optimal scalability while ILU(0) is not scalable in terms of iteration count with respect to the ratio $\frac{\Lambda_f}{\Lambda_m}$ between the fracture and matrix permeabilities. The same remark also holds in terms of scalability with respect to the mesh size which means that the ILU(0) preconditioner is only advantageous for small size and mildly heterogeneous problems.

Tables 1 and 2 also clearly show that the BiCGStab linear solver outperforms the GMRES linear solver for cases

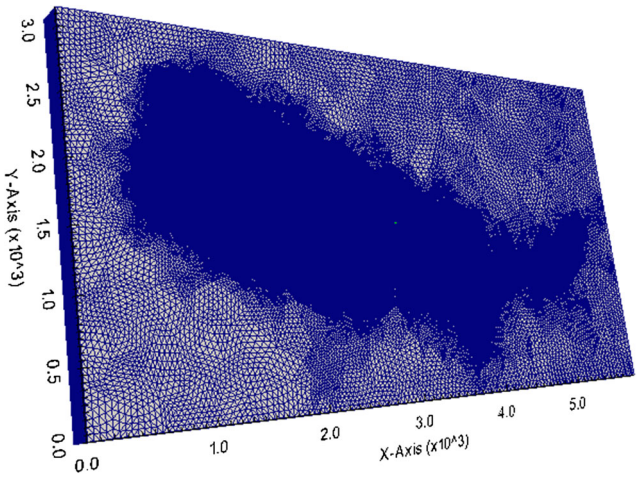


Fig. 17 Prismatic mesh of the domain Ω defined by the tensor product of a vertical 1D uniform mesh with a 2D triangular mesh

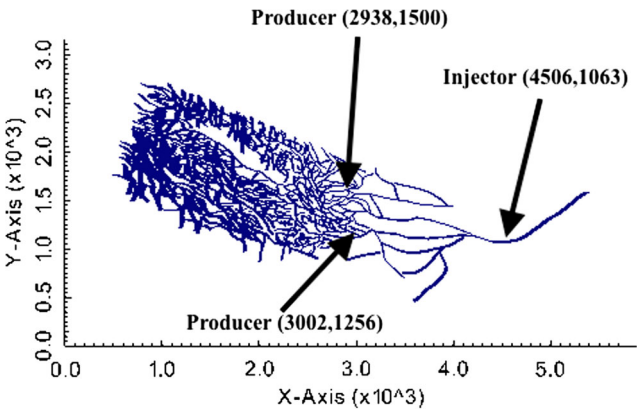


Fig. 18 Fracture network showing the location of the single injection well and of the two production wells

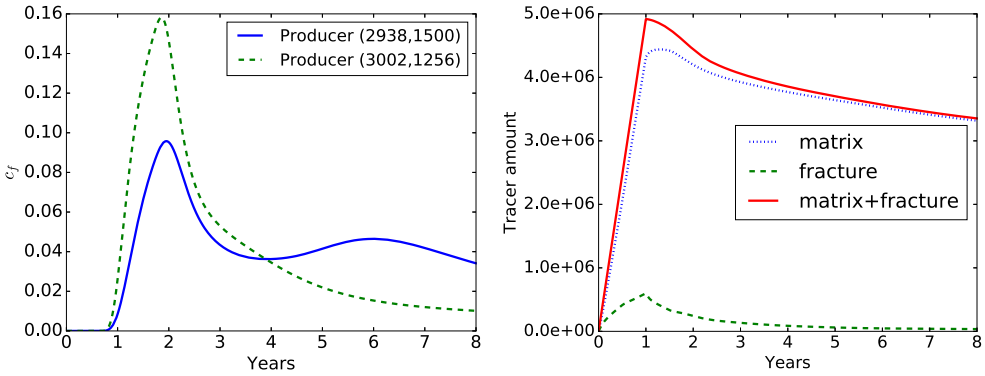


Fig. 19 Mean tracer concentration in both production wells as a function of time (*left*) defined as the ratio between the well tracer flow rate and the well fluid flow rate (equal in our case to the well fracture-face tracer concentration). Volume of tracer as a function of time in the matrix domain, in the fracture network and their sum (*right*)

requiring a large number of iterations due to the fact that the cost of the orthogonalization procedure increases with the dimension of the Krylov subspace. The Aggregation AMG preconditioner yields a larger number of iterations compared with the Boomer AMG preconditioner but has a much lower setup time resulting for this test case in a total lower CPU time. However, this implementation of the Aggregation AMG preconditioner seems to lack robustness with respect to the matrix fracture permeability ratio as exhibited in Table 3.

Next, Fig. 13 plots the total (Darcy flow and transport models) computation time and the computation time for the transport model only as a function of the number of processes. In these runs, the GMRES linear solver is used combined with the Boomer AMG preconditioner for $N_p \leqslant 128$ and with the ILU(0) preconditioner for $N_p = 256, 512$. For the range 2–512 of the number of processes, it appears that the computation time of the Darcy flow linear system solution remains small compared with the transport model computation time. This can be checked by comparison of the computation times in Table 2 and in Fig. 13. This explains the good scalability obtained for both the total and transport computation times thanks to the explicit nature of the time integration scheme.

5.4 Fracture network with tetrahedral meshes

This test case considers tetrahedral meshes of the cubic domain $(0, 1)^3$ conforming to the fracture network. An example of tetrahedral mesh showing both the matrix domain and the fracture network is exhibited in Fig. 14. All the physical parameters, initial and boundary conditions are the same as for the previous test case. The mesh used in this subsection contains about 6.2×10^6 cells, 9.7×10^5 nodes and 7.1×10^4 fracture faces. Figure 15 exhibits the tracer concentrations obtained with this tetrahedral mesh at times $t = 0$, $t = 0.2$, $t = 0.4$ and at the final simulation time $t_f = 0.5$.

As for the previous test case, Tables 4, 5, and 6 investigate the performance of the Darcy flow system linear solution for both the GMRES and BiCGStab linear solvers and for the same three preconditioners as in the previous test case. The conclusions are basically the same as for the hexahedral mesh test case. The Boomer AMG preconditioner exhibits an optimal robustness with respect to the matrix fracture permeability ratio $\frac{\Lambda_f}{\Lambda_m}$. On the other hand, it requires a rather high number of unknowns per MPI process to maintain a good parallel scalability due to the high level of communications in particular in the setup phase. The ILU(0) preconditioner can be an interesting alternative but only for small size and midly heterogeneous problems. The aggregation AMG preconditioner from the Trilinos library used in our test cases seems to lack robustness and we did not manage to make it work better through tuning its parameters.

Figure 16 plots the total (Darcy flow and transport models) computation time and the computation time of the transport model only as a function of the number of processes. In these runs, the GMRES linear solver is used combined with the Boomer AMG preconditioner for $N_p \leqslant 128$ and with the ILU(0) preconditioner for $N_p = 256, 512$. Compared with the previous subsection, an even better parallel scalability of the transport model computation time is observed in the right Fig. 16. This can be explained by the ratio of roughly 6 between the number of cells and the number of nodes typical of a tetrahedral mesh. For a topologically Cartesian mesh, this ratio is roughly 1. Since the cell concentrations are computed locally in each process,

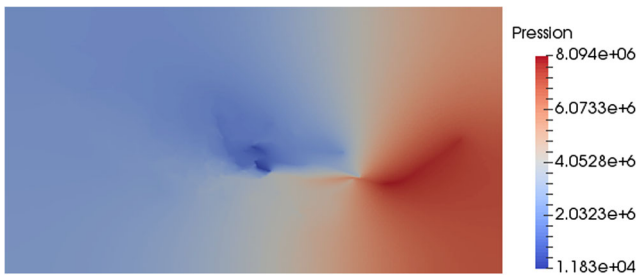


Fig. 20 Pressure on the matrix domain

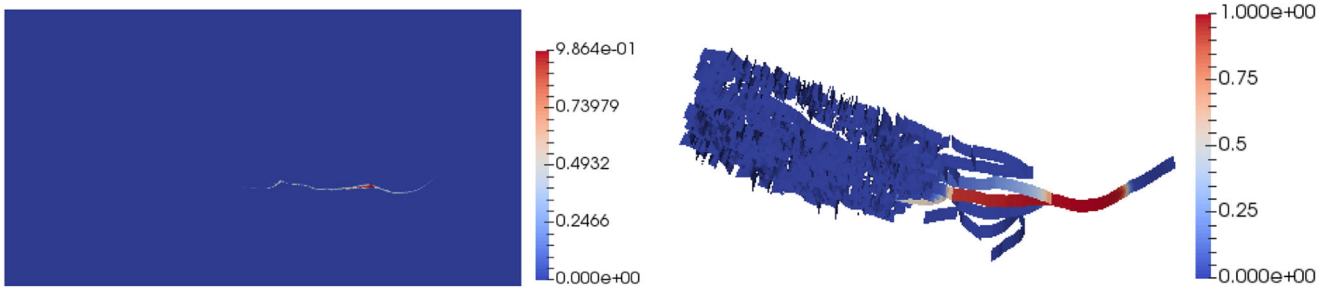


Fig. 21 Tracer concentration after 1 year of injection in the matrix domain (*left*) and in the fracture network (*right*)

this explains the better scalability observed for this tetrahedral mesh compared with the previous hexahedral mesh. On the left Fig. 16, it is observed that the linear system solution computation time is no longer small compared with the transport computation time for $N_p = 256$ and 512. Hence, it significantly reduces the parallel efficiency of the simulation for a large number of processes, say $N_p = 256, 512$ in this test case.

5.5 Application to a complex fracture network

In this subsection, our algorithm is applied to a complex fracture network kindly provided by M. Karimi-Fard and A. Lapène from Stanford University and TOTAL. Figure 17 exhibits the mesh of the domain $\Omega = (0, 5888.75) \times (0, 3157.5) \times (0, 250)$ (m) which contains about 1.2×10^7 prismatic cells, 6.5×10^6 nodes and 5.13×10^5 fracture faces. This 3D mesh is defined by the tensor product of a triangular 2D mesh with a uniform vertical 1D mesh with 24 intervals. The fracture network exhibited in Fig. 18 contains 581 connected components. It is a set of 21376 × 24 faces of the 3D mesh defined by the tensor product of a subset of 21,376 edges of the triangular 2D mesh with the 1D vertical mesh. The 2D triangular mesh contains 517,540 cells and is refined in the neighborhood of the fracture network down to an average size of 3.5 m. Figure 18 also shows the location of the injection well and of the two production wells. Each well is vertical of radius $r_w = 0.1$ m and its center in the horizontal plane is located at the middle of a fracture edge in the 2D triangular mesh. In the vertical direction, only the 12 fracture faces at the center of the 1D mesh are perforated. The permeabilities are isotropic and set to $\Lambda_f = 10^{-11}$ (m^2) in the fracture network and to $\Lambda_m = 10^{-15}$ (m^2) in the matrix domain. The porosities are set to $\phi_m = \phi_f = 0.1$, the fracture width to $d_f = 1$ m and the fluid viscosity to $\mu = 10^{-3}$ Pa s^{-1}.

The initial concentration is set to 0 both in the matrix domain and in the fracture network. A total volume of 5.0×10^6 m^3 is injected in one year at the injector well with a tracer concentration of 1. The pressures of each perforated fracture face σ of the producer wells are fixed to $p_w = 0$ and the flow rates are given by the Peaceman model

$$q_\sigma = WI_\sigma(p_\sigma - p_w),$$

where p_σ is the pressure in the fracture face and WI_σ the well index of the fracture face. This well index is computed following Peaceman methodology [14, 36, 37] by expanding the fracture face as a box of size $dx \times d_f \times dz$. The analytical pressure solution obtained for a vertical well with the well pressure p_w, the well radius r_w and the well flow rate q_w per unit length is imposed at the eight corners of the box. Then, the flow rate $q_w dz$ is imposed at the box

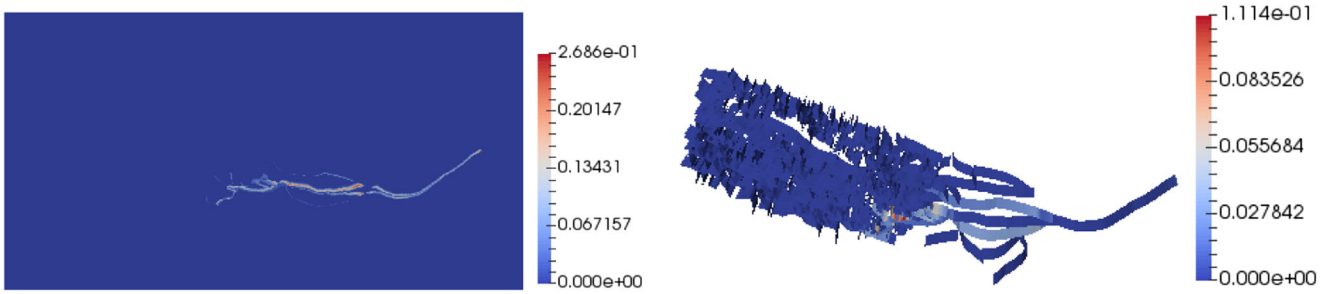


Fig. 22 Tracer concentration at final time in the matrix domain (*left*) and in the fracture network (*right*)

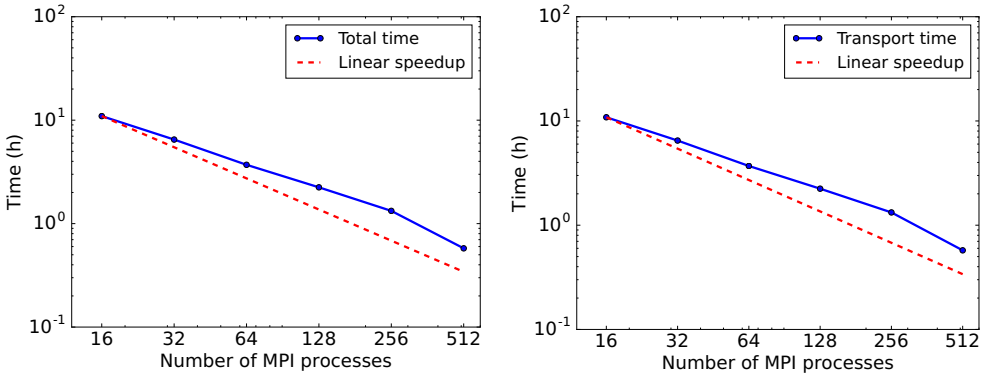


Fig. 23 Total computation time in hours (*left*) and computation time for transport model (*right*) vs. number of MPI processes with prismatic mesh

center and the pressure p_c at the box center can be computed analytically using the VAG scheme. We deduce the well index

$$WI = \frac{q_w dz}{p_c - p_w}$$

leading in this simple case to the analytical formula

$$WI = \frac{2\pi dz \Lambda_f}{\log\left(\frac{r_0}{r_w}\right)}$$

with

$$r_0 = D \exp\left(-\frac{2\pi dz}{C}\right),$$

and

$$C = \frac{4}{3}\left(\frac{dxdz}{d_f} + \frac{dxd_f}{dz} + \frac{dzd_f}{dx}\right), \quad D = 0.5\sqrt{dx^2 + d_f^2}.$$

The production lasts 8 years.

Figure 19 plots the mean tracer concentration in each well as a function of time as well as the total volume of tracer as a function of time in the matrix, in the fracture network and their sum. Figure 20 exhibits the pressure solution in the matrix domain and Figs. 21 and 22 show the tracer concentration after one year of injection and at final time both in the matrix domain and in the fracture network.

Figure 23 shows the total computation times with different number of MPI processes $N_p = 16, 32, 64, 128, 256, 512$. It is observed that the total computation time exhibits a rather good scalability. In addition, the linear solver (GMRES+Boomer AMG) for the pressure converges in no more than 25 iterations whatever the number of MPI processes. Also, the comparison of the total and transport computation times in Fig. 23 shows that the time for the pressure solution remains small compared with the transport computation time up to $N_p = 512$.

6 Conclusion

This paper introduced a parallel VAG scheme for the simulation of a hybrid dimensional Darcy flow and transport model in a discrete fracture network taking into account the mass exchanges with the matrix. The convergence of the scheme was validated on two original analytical solutions for a flow and transport model that includes fractures. The parallel efficiency of the algorithm was studied for different complexities of fracture networks, and a large range of matrix fracture permeability ratios and different type of meshes. The numerical results exhibit a very good parallel strong scalability as expected from the explicit nature of the time integration of the transport model with a better result on tetrahedral meshes thanks to the communication free computation of the cell unknowns. The Darcy flow solution is remarkably robust using the Boomer AMG preconditioner on all types of fracture networks, meshes, and for all permeability ratios that have been tested. On the other hand, it requires as usual a rather high number of unknowns per process to maintain a good parallel scalability. Future work includes the extension of the parallel algorithm to hybrid dimensional multiphase flow models and the use of a more accurate second order MUSCL scheme for the transport model.

Acknowledgments This work is supported by a joint project between INRIA and BRGM Carnot institutes (ANR, INRIA, BRGM). This work was also granted access to the HPC and visualization resources of "Centre de Calcul Interactif" hosted by University Nice Sophia Antipolis. This work was partially funded by ADEME BRGM research project Orbou (convention 1305C0131). We would like also to thank Mohammad Karimi-Fard and Alexandre Lapène for kindly providing us the mesh of Section 5.5.

References

1. Ahmed, R., Edwards, M.G., Lamine, S., Huisman, B.A.H., Pal, M.: Control-volume distributed multi-point flux approximation coupled with a lower-dimensional fracture model. J. Comput. Phys. **284**, 462–489 (2015)
2. Ahmed, R., Edwards, M.G., Lamine, S., Huisman, B.A.H., Pal, M.: Three-dimensional control-volume distributed multi-point flux approximation coupled with a lower-dimensional surface fracture model. J. Comput. Phys. **303**, 470–497 (2015)
3. Alboin, C., Jaffré, J., Roberts, J.E., Serres, C.: Modeling fractures as interfaces for flow and transport in porous media. Fluid flow and transport in porous media **295**, 13–24 (2002)
4. Amir, L., Kern, M., Martin, V., Roberts, J.E.: Décomposition de domaine et préconditionnement pour un modèle 3D en milieu poreux fracturé Proceeding of JANO 8. In: 8th conference on Numerical Analysis and Optimization (2005)
5. Angot, P., Boyer, F., Hubert, F.: Asymptotic and numerical modelling of flows in fractured porous media. ESAIM Math. Model. Numer. Anal. **23**, 239–275 (2009)
6. Antonietti, P.F., Formaggia, L., Scotti, A., Verani, M., Verzott, N.: Mimetic finite difference approximation of flows in fractured porous media. ESAIM Math. Model. Numer. Anal. **50**, 809–832 (2016)
7. Balay, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Zhang, H.: PETSc Users Manual. Revision **3**, 5 (2015)
8. Berrone, S., Pieraccini, S., Scialò, S.: An optimization approach for large scale simulations of discrete fracture network flows. J. Comput. Phys. **256**, 838–853 (2014)
9. Bogdanov, I.I., Mourzenko, V.V., Thovert, J.-F., Adler, P.M.: Two-phase flow through fractured porous media. Phys. Rev. E **68**, 026703 (2003)
10. Brenner, K., Masson, R.: Convergence of a vertex centered discretization of two-phase Darcy flows on general meshes. Int. J. Finite Vol. Methods (2013)
11. Brenner, K., Groza, M., Guichard, C., Masson, R.: Vertex approximate gradient scheme for hybrid dimensional two-phase Darcy flows in fractured porous media. ESAIM Math. Model. Numer. Anal. **49**, 303–330 (2015)
12. Brenner, K., Groza, M., Guichard, C., Lebeau, G., Masson, R.: Gradient discretization of hybrid dimensional Darcy flows in fractured porous media. Numer. Math., 1–41 (2015)
13. Brenner, K., Hennicker, J., Masson, R., Samier, P.: Gradient discretization of hybrid dimensional Darcy flows in fractured porous media with discontinuous pressure at matrix fracture interfaces. IMA J. Numer. Anal., published online (2016)
14. Chen, Z., Zhang, Y.: Well flow models for various numerical methods. Int. J. Numer. Anal. Model. **6**(3), 375–388 (2009)
15. Dalissier, E., Guichard, C., Havé, P., Masson, R., Yang, C.: COMPASS: a tool for distributed parallel finite volume discretizations on general unstructured polyhedral meshes. ESAIM: Proc. **43**, 147–163 (2013)
16. D'Angelo, C., Scotti, A.: A mixed finite element method for Darcy flow in fractured porous media with non-matching grids. ESAIM Math. Model. Numer. Anal. **46**(2), 465–489 (2012)
17. Droniou, J., Eymard, R., Gallouët, T., Herbin, R.: Gradient schemes: a generic framework for the discretisation of linear, nonlinear and nonlocal elliptic and parabolic equations. Math. Models Methods Appl. Sci. **23**(13), 2395–2432 (2013)
18. Droniou, J., Eymard, R., Gallouët, T., Guichard, C., Herbin, R.: The Gradient discretization method: a framework for the discretization of linear and nonlinear elliptic and parabolic problems. Preprint, https://hal.archives-ouvertes.fr/hal-01382358 (2016)
19. Eymard, R., Guichard, C., Herbin, R.: Small-stencil 3D schemes for diffusive flows in porous media. Math. Model. Numer. Anal. **46**, 265–290 (2010)
20. Eymard, R., Herbin, R., Guichard, C., Masson, R.: Vertex centred discretization of compositional multiphase Darcy flows on general meshes. Comput. Geosci. **16**(4), 987–1005 (2012)
21. Eymard, R., Guichard, C., Masson, R.: High performance computing linear algorithms for two-phase flow in porous media. In: FVCA 7 Proceedings (2014)
22. Flauraud, E., Nataf, F., Faille, I., Masson, R.: Domain Decomposition for an asymptotic geological fault modeling. C. R. Acad. Sci. Méc. **331**, 849–855 (2003)
23. Faille, I., Fumagalli, A., Jaffré, J., Roberts, J.E.: Model reduction and discretization using hybrid finite volumes of flow in porous media containing faults. Comput. Geosci. **20**, 317–339 (2016)
24. Formaggia, L., Fumagalli, A., Scotti, A., Ruffo, P.: A reduced model for Darcy's problem in networks of fractures. ESAIM Math. Model. Numer. Anal. **48**(4), 1089–1116 (2014)
25. Fumagalli, A., Scotti, A.: A reduced model for flow and transport in fractured porous media with non-matching grids. Numer. Math. Adv. Appl. **2011**, 499–507 (2013)
26. Haegland, H.: Streamline methods with application to flow and transport in fractured media. PhD thesis. University of Bergen (2009)
27. Heroux, M.A., Willenbring, J.M.: Trilinos Users Guide (2003)
28. Geiger, S., Huangfu, Q., Reid, F., Matthai, S., Coumou, D., Belayneh, M., Fricke, C., Schmid, K.: Massively parallel sector scale discrete fracture and matrix simulation. Society of Petroleum Engineers (2009)
29. Hoteit, J., Firoozabadi, A.: An efficient numerical model for incompressible two-phase flow in fracture media. Adv. Water Resour. **31**, 891–905 (2008)
30. Hypre - Parallel high performance preconditioners, http://acts.nersc.gov/hypre
31. Karimi-Fard, M., Durlofsky, L.J., Aziz, K.: An efficient discrete-fracture model applicable for general-purpose reservoir simulators. Society of Petroleum Engineers (2004)
32. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**(1), 359–392 (1998)
33. Martin, V., Jaffré, J., Roberts, J.E.: Modeling fractures and barriers as interfaces for flow in porous media. SIAM J. Sci. Comput. **26**(5), 1667–1691 (2005)
34. Matthai, S.K., Mezentsev, M.A., Belayneh, M.A.: Finite element-node-centred finite-volume two-phase-flow experiments with fractured rock represented by hybrid-element. SPE Reserv. Eval. Eng. **12**, 740–756 (2007)
35. Monteagudu, J., Firoozabadi, A.: Control-volume model for simulation of water injection in fractured media: incorporating matrix heterogeneity and reservoir wettability effects. SPE J. **12**(03), 355–366 (2007)
36. Peaceman, D.W.: Interpretation of well-block pressures in numerical reservoir simulation. SPE J. **18**(03), 183–94 (1978)
37. Peaceman, D.W.: Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. SPE J. **23**(03), 531–43 (1983)

38. Reichenberger, V., Jakobs, H., Bastian, P., Helmig, R.: A mixed-dimensional finite volume method for multiphase flow in fractured porous media. Adv. Water Resour. **29**(7), 1020–1036 (2006)
39. Sandve, T.H., Berre, I., Nordbotten, J.M.: An efficient multi-point flux approximation method for discrete fracture-matrix simulations. J. Comput. Phys. **231**, 3784–3800 (2012)
40. Schwenck, N., Flemisch, B., Helmig, R., Wohlmuth, B.: Dimensionally reduced flow models in fractured porous media: crossings and boundaries. Comput. Geosci. **19**, 1219–1230 (2015)
41. Si, H.: http://tetgen.org
42. Tunc, X., Faille, I., Gallouët, T., Cacas, M.C., Havé, P.: A model for conductive faults with non matching grids. Comput. Geosci. **16**, 277–296 (2012)