Advances in computation of local problems for a flow-based upscaling in fractured reservoirs

Alessio Fumagalli[♯] Stefano Zonca[♯] Luca Formaggia[♯]

October 30, 2015

[♯] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
alessio.fumagalli@polimi.it
stefano.zonca@polimi.it
luca.formaggia@polimi.it

Abstract

In this article we present some advances to increase the efficiency and applicability of a flow-based upscaling procedure to solve single and multiphase flows in natural fractured reservoirs. These geological formations may be characterized by hundreds up to hundreds of thousands of fractures, ranging from small to medium scales, which spread all the reservoir. An explicit representation of all the fractures in real scenarios make soon unfeasible performing numerical simulations. An upscaling procedure is thus required. We assume that the reservoir can be modeled with a coarse corner-point grid where the fractures are geometrically uncoupled, by using an embedded discrete fracture model. To describe the scaled up problem, we consider a flow-based upscaling procedure where multiple sub-regions are used to derive transmissibilities, mean depths and pore volumes related to the coarse degrees of freedom associated with fractures and rock matrix. Our focus is to further enhance the upscaling process by allowing the splitting of unconnected rock matrix regions and to compare two ways of setting up the local problems used to compute the transmissibility between coarse cells. Numerical examples confirm the effectiveness of the proposed approach.

1 Introduction

Natural fractured reservoirs are one of the most challenging problem for the extraction and exploitation of hydrocarbons trapped in the underground, mainly

in the rock matrix [2]. On top of the classical difficulties, *e.g.* lack of data and uncertainty, a new difficulty arises in these geological formations which is the presence of a large number of fractures. Their thickness is orders of magnitude smaller than their other characteristic sizes, while their permeability is several orders of magnitude higher than the surrounding porous matrix, therefore the flow of liquid and gas phases tend to flow along them. To obtain an accurate numerical solution it is essential to include fractures in the mathematical model overcoming their intrinsic difficulties. In this work, we suppose that the fractures are fixed object known a priori and the rock matrix is described by a corner-point grid, normally used in the oil industry, which is independent of the fractures.

Real problems, involving hundreds of thousands of fractures with rock meshes composed by thousands of cells, are unaffordable when considering standard approaches where a grid is constructed honouring the geometry of all the fractures. We choose instead an upscaling strategy to describe the global problem at a scale-up or coarse level, defined on the rock grid, where the behaviour of the fractures and the surrounding matrix are described via coarse parameters, as explained in [6]. The authors in [13, 9, 11, 10, 19] introduced a systematic flow-based upscaling algorithm that produces, from the solution of local problems involving single or pair of coarse cells, a generalized Dual-Porosity/Dual-Permeability (DPDK) model. This methodology, called Multiple Sub-Region (MSR), introduces local sub-regions or portions of local coarse cells to enhance the description of the dynamics within the porous medium. The solution of the local problems aims at deriving effective coarse-scale parameters for describing matrix-matrix, fracture-matrix and fracture-fracture interaction. The MSR method gives a better characterization of the flux exchange, inside each coarse block, between the fracture network and the rock matrix since the definition of the sub-regions reflects the real fracture distribution. The computed properties, *i.e.* transmissibilities between coarse degrees of freedom, mean depths and pore volumes of the coarse degrees of freedom, form the upscaled description of the global problem. These parameters can be used to perform a single-phase or a multi-phase flow simulation by means of a reservoir simulator.

For the solution of the local problems the Discrete Fracture Model (DFM) represents one of the most accurate methodology for describing flow in fracture networks. However, the DFM is still too computational demanding since it requires a conforming mesh among the fractures and the rock matrix, which can be composed of many elements and of poor quality, can be costly. We consider an alternative model called Embedded Discrete Fracture Model (EDFM), see [14, 18, 16, 5], which represents an optimal compromise between accuracy and efficiency by enabling the possibility to adopt standard corner-point grids for the background matrix together with a non-matching representation among the fractures and the rock matrix itself.

In this work we present some advances for the standard MSR upscaling procedure. Our main focus is to enhance the rock matrix description, at the coarse level, introducing a splitting of the unconnected parts for each sub-region. We

obtain a self adapt algorithm where the level of details is automatically chosen from the fracture distribution. In fact, more intersected fractures normally generate more unconnected parts of the same sub-region. Moreover, we present and compare two possible alternatives for the problem used to compute the upscaled transmissibility between facing coarse cells. One possibility, see [9], is to impose a pressure jump at the boundaries "parallel" to the interface, while the other introduced for unfractured reservoir, assumes a linear pressure profile on all the boundaries of the pair of cells. Advantages and comparisons of the proposed techniques will be discussed by presenting numerical examples.

This paper is organized as follow: in Section 2, we recall the upscaling procedure where EDFM is used for the representation of the fractures. Section 3 describes the construction of single coarse cell problems by introducing the algorithm used to split the rock matrix sub-regions. In Section 4, we present two possibilities to compute the upscaled transmissibility between coarse cells. Section 5 presents academic test cases as well as a 3D scenario of a natural fractured reservoir to make a comparison of the proposed methods. Finally, Section 6 is devoted to the conclusions.

2 Upscaling in fractured reservoir

In this section we present a flow-based upscaling procedure to derive effective parameters at a coarse level. The basic idea is to lump the properties defined in the fine scale in order to obtain a representative value for each coarse block. These parameters are computed by solving local single-phase flow problems and subsequently used for a two-phase flow simulation. This choice, even if not completely satisfactory, is usually accepted to speed up the overall process. Moreover the proposed procedure may be advantageous in multiple scenario simulations in which it is needed to repeat the two-phase flow simulation in several settings, *e.g.* considering different boundary conditions, number of wells and their location.

2.1 A model for fractured reservoirs

Let $\Omega \in \mathbb{R}^3$ be the physical domain that represents the reservoir. We assume that Ω be a regular domain with Lipschitz boundary, denoted by $\partial\Omega$. The Darcy's law coupled with the mass conservation equation may be used to model a fluid flow through a porous medium:

$$\begin{cases} \boldsymbol{u} = -\dfrac{\Lambda}{\mu}\nabla p \\ \nabla \cdot \boldsymbol{u} = f \end{cases} \quad \text{in } \Omega, \tag{1}$$

where $\boldsymbol{u}$ is the flow velocity, p the pressure, Λ is the symmetric positive definite permeability tensor, μ the dynamic viscosity and f is the volumetric source/sink

term. We have assumed that the fluid and the medium are incompressible and we also have neglected the gravitational effects.

Let us now consider a fracture γ that divides the domain Ω into two parts, Ω_1 and Ω_2 such that $\overline{\Omega} = \overline{\Omega_1} \cup \overline{\Omega_2}$, and let us introduce the outward normal from the domain Ω_i, $\boldsymbol{n}_i$, with $i = 1, 2$. Since the fracture has an aperture (thickness) d of several orders of magnitude smaller than its other characteristic sizes and than the characteristic sizes of the reservoir, it may be modeled as an object of co-dimension 1. For this reason, we describe the flow through a fracture γ with the reduced model presented in [15]:

$$\begin{cases} \hat{\boldsymbol{u}} = -\dfrac{\lambda_\tau}{\mu} d \nabla_\tau \hat{p} \\ \nabla_\tau \cdot \hat{\boldsymbol{u}} = \hat{q} + (\boldsymbol{u}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{u}_2 \cdot \boldsymbol{n}_2) \end{cases} \quad \text{on } \gamma, \tag{2}$$

where we have denoted with $\hat{\cdot}$ the quantities defined on the fracture γ, $\nabla_\tau \cdot$ and ∇_τ denote the tangential divergence and tangential gradient operator, respectively, λ_τ is the permeability along the fracture and $\boldsymbol{u}_i$ with $i = 1, 2$ is the restriction of the Darcy velocity on Ω on the two sides of the fracture. To couple the porous medium Ω with the fracture γ, we consider the following conditions

$$\begin{cases} \boldsymbol{u}_1 \cdot \boldsymbol{n}_1 = 2\dfrac{\lambda_n}{d}(p_1 - \hat{p}) \\ \boldsymbol{u}_2 \cdot \boldsymbol{n}_2 = 2\dfrac{\lambda_n}{d}(p_2 - \hat{p}) \end{cases} \quad \text{on } \gamma, \tag{3}$$

where λ_n is the permeability across the fracture and where p_i with $i = 1, 2$ is the restriction of the pressure on Ω on the two side of the fracture. If a fracture does not cross entirely the domain a no flow condition is imposed on the internal boundaries. Moreover in the case of intersecting fractures the pressure continuity and conservation of mass is prescribed at the intersection, see [3]. The problem described by equations (1),(2) and (3) can be proved to be equivalent to a problem where equations (1) are used to model both the porous medium and the fractures, by treating the latter as virtual cells. Each virtual cell is the widening, in the normal direction, of a fracture by the thickness d. The interested reader can refer to [7, 8] for a detailed description of this method. To keep the notation as simple as possible in the forthcoming sections we make use of this equivalence to let the differential problem in the fractures be implicit.

2.2 The upscaling procedure

In this part we briefly present the flow-based upscaling procedure introduced in [12], see also [9, 11, 10, 19]. There are two main ingredients: a coarse mesh, where the scale-up variables will be defined, and the fractures which are independently disposed in the coarse mesh. At the upscaled level, in each coarse block, the fractures are no longer explicitly represented but all of them are described by a single degree of freedom. Moreover, in each coarse block, it is possible

to introduce several rock matrix degrees of freedom associated with a suitable portion of the bulk in the coarse cell. The idea is to have, at a coarse level, a scheme which is equivalent to the well-known two-point flux approximation where the quantities involved are not computed from geometrical properties but derived from local computations. These *d.o.f.*s are linked each other by the upscaled transmissibilities, which is the main feature in the scheme, as well as each *d.o.f.* is characterised by mean depth and pore volume. These coarse scale quantities are computed by solving local problems which involve single and pairs of coarse cells. The solution of a differential problem on a single coarse cells is used to define how to split the contained matrix rock, by using the multiple sub-regions algorithm explained in Section 3, and then to compute the upscaled transmissibilities among the matrix parts and fracture *d.o.f.* . Moreover, for each of these problems, the mean depths and the total pore volumes are computed for each local degree of freedom. Once the single cell problems are processed, the pairs of coarse cells are considered and we set up a differential problem used to derive the interchange, *i.e.* transmissibility, among matrix and fracture *d.o.f.* of one coarse cell to the corresponding matrix and fracture *d.o.f.* of the other cell. We present this problem in Section 4. Since this procedure requires the solution of several local problems where tens or even hundreds of fractures are involved, we consider an efficient and effective numerical scheme to speed up their computation. To overcome the difficulties related to the grid generation, we allow the fractures to be uncoupled among each other and the mesh of the rock matrix. We consider a non-matching representation of the fractures, see [14, 18, 16, 17, 5], by using the embedded discrete fracture method (EDFM) where the matrix corresponds to a background discretization. In the aforementioned works the authors consider the fractures immersed in the matrix grid. Moreover the meshes of the fractures are constructed based on the rock mesh, *i.e.* each fracture cell correspond univocally to a matrix cell. Since the fractures are considered to be highly conductive compared to the surrounding medium, approximated formulae can be derived to couple the fractures cells and the matrix cells. We assume linear fractures composed by four points lying on a plane and, since we are considering corner-point grids, the faces of the rock matrix are bilinear as presented in [18, 17]. We consider also each fracture geometrically independent with respect to the others, then suitable coupling conditions which describe the linking among fractures cells can be derived. In this way, since we are considering a coarse corner-point grid, the construction of the computational mesh associated with the rock matrix is trivial and can be performed with almost no computational overhead.

3 Single coarse cell

In this section, we show how to solve the local problem within each single coarse cell. Since a single coarse cell can be seen as a small reservoir from which it

is possible to produce oil, it is reasonable to consider, at a first approximation, each cell completely isolated to the others and hence, to treat one cell at a time, whose exchanges of flow are just between the porous matrix and the fractures contained in the cell itself. The basic idea is to solve a time-dependent problem until a pseudo-steady state is reached and then to subdivide the coarse cell into blocks (or sub-regions), so that it is possible to compute several upscaled transmissibilities among these blocks. Following [13], we consider the compressible single-phase flow equation defined on a generic cell K

$$\begin{cases} c\phi\dfrac{\partial p}{\partial t} - \nabla\cdot\dfrac{\Lambda}{\mu}\nabla p = f & \text{in } K \text{ for } t>0 \\ -\dfrac{\Lambda}{\mu}\nabla p\cdot\boldsymbol{n}_K = 0 & \text{on } \partial K \text{ for } t>0 \\ p = 0 & \text{in } K \text{ for } t=0 \end{cases}, \tag{4}$$

where ∂K is the boundary of the coarse cell K, $\boldsymbol{n}_K$ represents the outward normal to K, c is the compressibility, ϕ is the porosity and f is a strictly positive and constant source term that acts on the fractures and it is zero in the porous medium. A sketch of this setting is reported in Figure 1. The problem (4) is

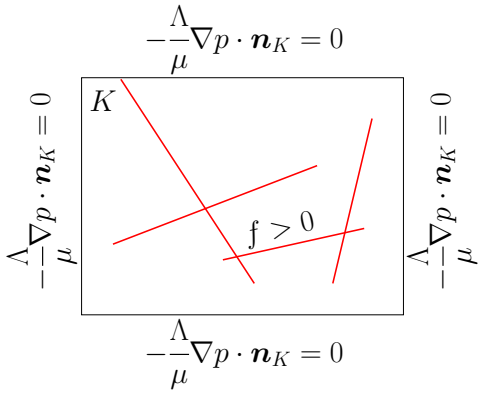

Figure 1: Schematic example of the local problem used to compute the upscaled transmissibilities.

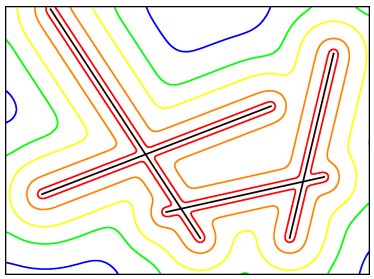

Figure 2: Schematic example of the isocontours of the pressure.

numerically solved until the following condition is satisfied

$$\frac{\partial p}{\partial t} \simeq const.$$

By indicating with $\bar{t}$ the time from which this condition is fulfilled, for all $t \geq \bar{t}$, the solution is said to be at the pseudo-steady state. It means that the dynamics inside the coarse cell does no longer change.

From the implementative viewpoint, problem (4) is replaced by an equivalent stationary version that avoids the temporal loop and hence saves computational

time. For details, see [8]. Once problem (4) is solved, we generate the sub-regions by considering the isocontours of the pressure solution, see Figure 2. In particular, for the cell K, we define S_K sub-regions where the first one, S_1, is reserved for the fractures and the other ones, S_i with $i = 2, \ldots, S_K$, are taken for the porous medium. These sub-regions are determined such that their boundaries are the pressure isocontours. These isocontours can be selected in several ways, in particular we decide to choose them such that the pressure range of the numerical solution is uniformly divided.

At this point, it is possible to compute the properties required by the up-scaling procedure. For each sub-region, fractures included, we compute the pore volume, *i.e.* the volume of the sub-region multiplied by the porosity, and the average depth of the block. Moreover, for each pair of adjacent sub-regions we compute the transmissibility as

$$T_{S_i,S_j} = \frac{Q_{S_i,S_j}}{p_{S_i} - p_{S_j}},$$

where the subscript i of S_i, respectively j, indicates that the quantity is related to the i-th sub-region, with $i = 2, \ldots, S_K$, T and Q are the transmissibility and the flux between the sub-regions, respectively, and p is the average pressure in the sub-region. The same formula is used to compute the transmissibility between the fractures and the sub-region that contains them. It may happen that the fractures may be contained into several sub-regions, so we have a separated contribution of transmissibility for each of them.

Remark 1. *In order to solve problem* (4) *in cell* K *and to compute the sub-regions, we create a local fine grid just for the calculation of the required quantities. There is no need to store this fine mesh, since it has only a temporary use.*

Remark 2. *To be more general we relax the assumption that each single coarse cells is isolated from the others and the communication takes place only through the fractures. The aforementioned approach is still valid but in the forthcoming section we consider also the matrix-matrix connection between facing coarse cells.*

3.1 Splitting unconnected sub-regions

The arrangement of the fractures inside the coarse cells, *i.e.* their position and their connection, may lead to configurations such that a matrix sub-region S_i may be composed of several unconnected parts, see an example in Figure 3, left. This situation may produce undesired results, since we assign the same degree of freedom for each unconneted part of the sub-region even though it is composed of unrelated portion of the domain. So, it is more reasonable to assign a different degree of freedom for each unconnected part of a sub-region. To do this, after the computation of the S_K (standard) sub-regions as described in the previous section, we simply apply the procedure described in Algorithm 1 so

that each portion of the coarse cell K has assigned a different degree of freedom, or colour, *i.e.* each portion represents a different sub-region. Algorithm 1 starts by considering a fine cell of the coarse cell K and recolours only the connected region on which this cell belongs to. Once this region is recoloured, the algorithm considers a different fine cell not yet recoloured, and then it recolours only the connected portion that contains this fine cell. The algorithm loops until all the fine cells are recoloured. Figure 3, right, shows the sub-regions after the splitting algorithm applied to the case in Figure 3, left. In this case, we consider that two fine cells are neighbours if they share a common face.

Algorithm 1 Splitting of unconnected sub-regions

Require: standard sub-regions
Ensure: split sub-regions

```
1:  markedCells = 0
2:  marker = # standard sub-regions
3:  q = ∅ {queue of cells to be processed}
4:  v = ∅ {list of marked cells}
5:  p = ∅ {list of processed cells}
6:  while markedCells < # cells do
7:     marker += 1
8:     c = findFirstUnmarkedCell()
9:     q.push(c)
10:    p.insert(c)
11:    oldColor = color(c)
12:    while q not empty do
13:       c = q.pop()
14:       q.remove(c)
15:       color(c) = marker
16:       v.insert(c)
17:       markedCell += 1
18:       for all neighbor cells n of cell c do
19:          if n ∉ v and n ∉ p and color(n) = oldColor then
20:             q.push(n)
21:             p.insert(n)
22:          end if
23:       end for
24:    end while
25: end while
```

We remark that by employing this procedure, it is not possible to know a priori the final number of sub-regions in the coarse cell. We will give a comparison of the classical and the splitting procedure in Section 5.1.

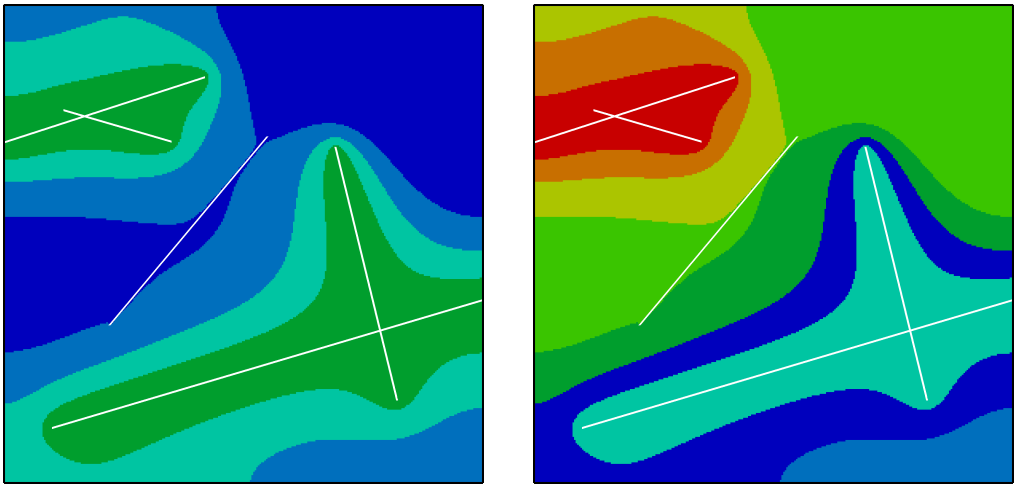

Figure 3: (Left) Classical subdivision of the coarse cell K into sub-regions, with $S_K = 5$ blocks, 1 for the fractures and 4 for the porous matrix. (Right) Subdivision of the same coarse cell K by applying the splitting algorithm. The original 4 matrix sub-regions are split for a total of 8 different matrix sub-regions.

4 Pair of coarse cells

In this section, we present and analyse two approaches to compute the upscaled transmissibility between facing coarse cells. There are several possibilities available in the literature, *e.g.* see [6, 9, 20], but many of them are focused on unfractured reservoirs. To ease the presentation, we consider a bi-dimensional framework however the extension to three-dimensional case is trivial. We consider the computational domain composed by two coarse cells K and L, with centres of mass $\boldsymbol{x}_K$ and $\boldsymbol{x}_L$, such that the differential problem is defined on $\overline{P} = \overline{K} \cup \overline{L}$ with $K \cap L = \emptyset$. We define the boundary of P as $\Gamma = \partial P$ and, since we consider our upscaled problem defined on a corner-point grid, we divide it into several disconnected parts: Γ^{top}, Γ^{bottom}, Γ^{left} and Γ^{right} such that $\overline{\Gamma} = \overline{\Gamma}^{\text{top}} \cup \overline{\Gamma}^{\text{bottom}} \cup \overline{\Gamma}^{\text{left}} \cup \overline{\Gamma}^{\text{right}}$, see Figure 4 for an example. The case of pinched edges can be easily handled but it is out of the scope of this presentation. The unit normal of Γ, pointing outward form P, is indicated with $\boldsymbol{n}_\Gamma$. Finally, we introduce the interface between the two coarse cells as $\overline{I} = \overline{K} \cap \overline{L}$. To solve the local problem, which will be presented in the sequel, we make use of the finite volume method with the standard two-point flux approximation (TPFA) scheme. Once the pressure profile is computed in P, following [9, 13], we can derive the upscaled transmissibility between the coarse cells K and L as

$$T_{KL} = \frac{F_I}{p_K - p_L},$$

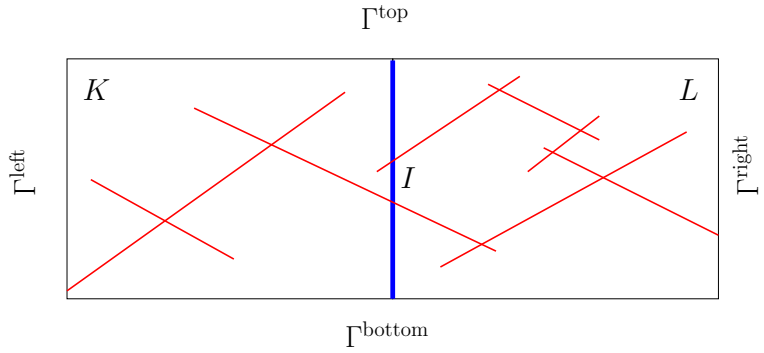

Figure 4: Example of pair of coarse cells with several intersecting fractures.

where F_I is the fractures flux through the interface I and p_K, p_L are the mean pressures in the fractures contained in the coarse cells K and L, respectively. The averages are computed on the fracture networks that cut the interface. In the TPFA scheme the transmissibility T is derived from geometrical properties and the permeability, by assuming a linear pressure profile inside the cells. Supposing that the permeability and the viscosity are uniform, we have

$$\int_I \frac{\Lambda}{\mu} \nabla p \cdot \boldsymbol{n}_I \mathrm{d}\boldsymbol{\sigma} = \frac{T_{KL}}{\mu} \left(p(\boldsymbol{x}_L) - p(\boldsymbol{x}_K) \right)$$

which is consistent for K-orthogonal grids, see [1]. We consider the same idea to derive a representative parameter for the transmissibility between K and L which mimics the formula derived for the two-points flux approximation. We start by considering the following problem without any volumetric source term

$$\begin{cases} -\nabla \cdot \dfrac{\Lambda}{\mu} \nabla p = 0 & \text{in } P \\ + \text{ boundary condition} & \text{on } \Gamma \end{cases} . \tag{5}$$

The main ingredient, among all the possible choices, is to define proper boundary conditions of problem (5) which give a linear pressure profile inside the two cells. The standard approach used in the MSR method, usually called "close boundary conditions", is to impose a pressure gradient on the two boundaries opposite at the interface I. By considering the division of Γ presented in Figure 4, the related boundary conditions are

$$\begin{cases} -\dfrac{\Lambda}{\mu} \nabla p \cdot \boldsymbol{n}_\Gamma = 0 & \text{on } \Gamma^{\text{top}} \cup \Gamma^{\text{bottom}} \\ p = p^{\text{left}} & \text{on } \Gamma^{\text{left}} \\ p = p^{\text{right}} & \text{on } \Gamma^{\text{right}} \end{cases} , \tag{6}$$

with $p^{\text{left}} \neq p^{\text{right}}$ and both constant on their respective boundary. This method gives linear pressure profiles for nearly Cartesian cells, but can behave poorly

for highly distorted cells or in the case where the fractures do not entirely cut the domain from Γ^{left} to Γ^{right}. In these cases the pressure profile may be far to be linear. We will see this behaviour in the Subsection 5.2. To force the solution to be more linear as possible inside the pair of cells, a possibility is to consider a linear pressure profile for all the boundaries of the pair of cells. This approach is known as "open boundary conditions". Considering $\alpha \in \mathbb{R} \setminus \{0\}$ and $\beta \in \mathbb{R}$ we impose on the boundary of P the following pressure

$$p(\boldsymbol{x}) = \alpha \boldsymbol{x} \cdot (\boldsymbol{x}_K - \boldsymbol{x}_L) + \beta \quad \text{on } \Gamma. \tag{7}$$

In this case the influence of the boundary conditions is higher and generally it gives a more linear pressure profile. The term $\boldsymbol{x}_K - \boldsymbol{x}_L$ is introduced to handle distorted cells, while α and β are additional parameters that can tune the solution to be in a certain range of values. The parameters presented in the previous cases do not affect the value of the coarse transmissibility but may help to solve the linear system associated with the discrete problems. In addition to the upscaled transmissibility associated with pairs of fracture coarse *d.o.f.* , we compute also the transmissibility among matrix sub-regions across the interface of the two coarse cells. These computations are independent to the choice of the boundary conditions and the split algorithm.

5 Numerical examples

In this section we present some test cases to validate the proposed approaches. First, we make a comparison of a two-phase flow solution computed by means of the upscaled parameters with and without using the split algorithm. In the second test case we compare the results obtained when the open or close boundary conditions are used for the local problems. Finally a 3D test case with a complex fracture network is analysed by employing the proposed methods.

5.1 Splitting of the sub-regions

In this test case we address the problem presented in Section 3.1 where unconnected matrix sub-regions can be split or not. We consider the 2D domain depicted in Figure 5 where a 6×6 Cartesian coarse mesh is represented as well as 70 intersecting fractures. The sizes of the reservoir are $304.8m \times 304.8m$ in the plane (x, y), thus we neglect the gravity force. The fractures thickness is $3.44cm$ and the ratio between fracture and rock permeability is four orders of magnitude, $10^6 mD$ and $100mD$ respectively. The rock porosity is 0.15 and the fractures porosity is 0.85. In each coarse block we consider a local mesh composed of 30×30-Cartesian fine cells. Figure 6 shows the sub-regions, coloured by their local number, for each coarse cell in the case without and with the split algorithm. In the first case we impose three matrix sub-regions for each coarse block while in the other case the number of matrix sub-regions ranges from three

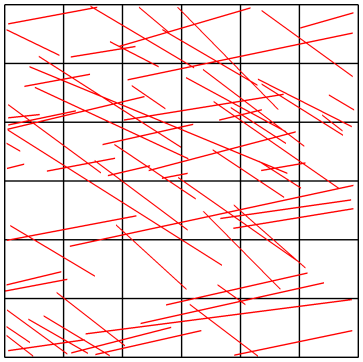

Figure 5: Coarse mesh with fractures for test case of Subsection 5.1.

up to twelve. Globally we obtain 144 degrees of freedom for the un-split case and 242 degrees of freedom for the split case. Moreover we have noticed that

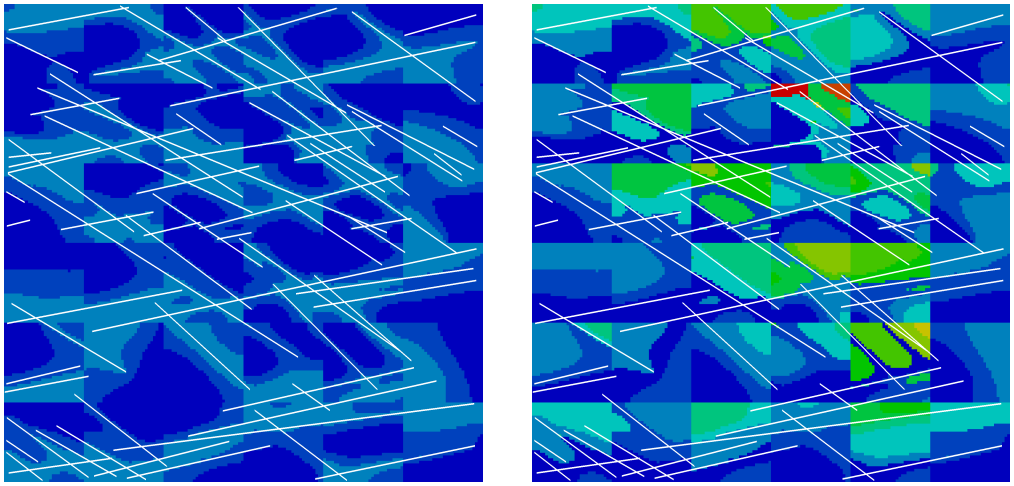

Figure 6: Sub-regions coloured by their local number for the un-split and split case. The two images are scaled by the same value.

the cost of the algorithm used to split the unconnected sub-regions is negligible, so that the cost of the upscaling procedure in the two cases is comparable.

Once the upscaled computation is performed, the transmissibilities and the pore volumes are used in a two-phase flow simulator. We have used the general purpose reservoir simulator (GPRS), developed by Stanford university [4], to compute the oil and water dynamics in the reservoir. In the two-phase flow simulation we consider the reservoir saturated by oil (density equal to 50 lbm/ft^3) with an injection well that pumps water (density equal to 64.79 lbm/ft^3) in the top right coarse cell and a production well in the bottom left coarse cell. Both wells are completed in the degree of freedom associated with the fractures of

the corresponding coarse cell. The production well operates at a fixed bottom-hole pressure (BHP) equal to $100psi$ and the injection well at a fixed water flow rate equal to $1.5 \cdot 10^5$ stb/day. The rock compressibility is set to $10^{-6}psi^{-1}$. Moreover, we neglect the residual saturation for both phases and the capillary pressure, and we assume a simple linear relation for the relative permeability. The simulation reproduces 3650 days. Figure 7 compares the oil saturation in the matrix sub-regions at the final time step for the un-split and split case. We

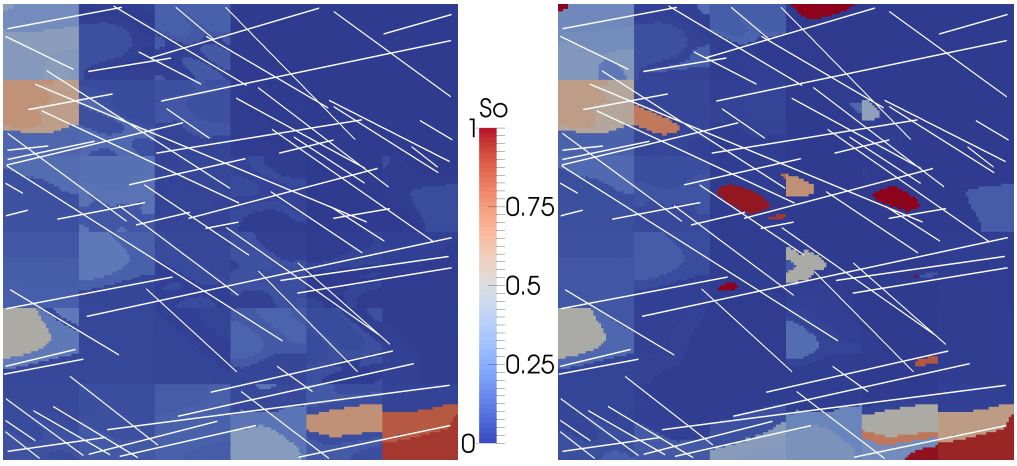

Figure 7: Solution in each matrix sub-region for the same time step using the un-split and split procedure.

clearly see that the two solutions are in good agreement except in few sub-regions in the centre and in the bottom right part of the domain. The reason is that in the split case some sub-regions may be more isolated, then the dynamics in these parts is slower. However, the behaviour of this solution is more physical than the solution obtained in the un-split case. In Figure 8 a comparison of the reservoir productions through the well in the two cases is presented. We notice that the production in the split case is slightly higher during the simulation, however at the end of the simulation we obtain the same value. Figure 9 shows the water cut, which is the ratio of water produced from the reservoir compared to the volume of total liquids, *i.e.* oil and water, produced. Also in this case the two algorithms behaves similarly with a delay in the split case. In both cases the steep variation in the first few days is due to the presence of the fractures, while the dynamics in the rock matrix is more smooth.

From this test, we deduce that by considering global indices, such as the amount of production or the water-cut, the splitting of the sub-regions does not produce qualitatively changes. However, a finer index, in particular the local saturation, is able to detect the differences between the two approaches and suggests that the splitting approach gives more accurate results.

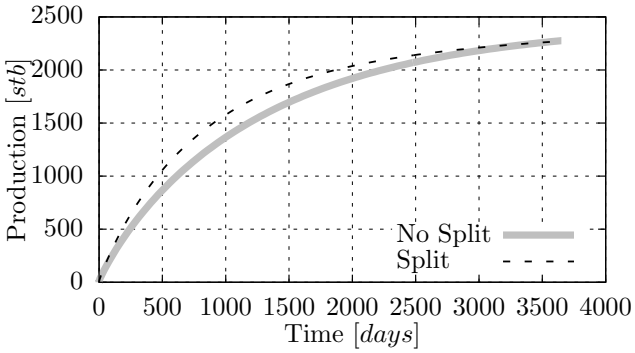

Figure 8: Comparison of the amount of oil produced from the well between the split and un-split approach.

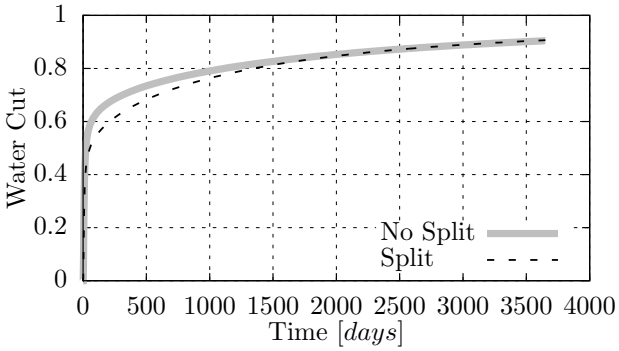

Figure 9: Comparison of the water cut between the split and un-split approach.

Remark 3. *By using the splitting algorithm, it is possible to create several sub-regions composed by one single fine cell. In these cases, the two-phase flow simulation may be hampered due to the ill-conditioning of the linear system introduced by the presence of these small sub-regions. For this reason, it may be necessary to consider a finer mesh for each coarse block in order to regroup these sub-regions. This can affect the computational cost of the upscaling procedure but lighten the two-phase flow simulation. However a parallelization can speed up the upscaling software. In the example presented above, each sub-region is composed by more than one fine cell.*

5.2 Boundary conditions

In this part we compare a two-phase flow solution computed with the upscaled parameters by using the open and the close boundary conditions (BC) to solve the differential problems associated with the pairs of coarse cells. We refer to Section 4 for the explanation of these methodologies. We consider the same fractures distribution, physical domain and properties as the previous test case. The coarse grid is defined by 6×6 cells, however, in this case they are no more Cartesian but distorted, and the fine grid generated in each coarse cell is composed of 100×100 cells. These changes are applied to emphasize possible differences between the approaches. See Figure 10 for a schematic representation of the computational domain. In this test case we consider only one matrix sub-region, for a total of 72 degrees of freedom. In Figure 11, we compare

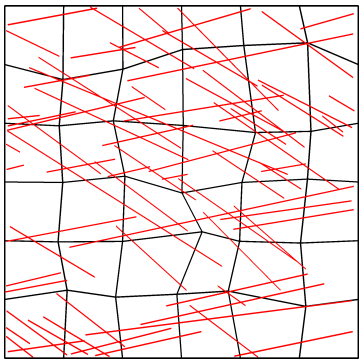

Figure 10: Coarse mesh with fractures for test case of Subsection 5.2.

the transmissibilities obtained in with the open BC with respect to the one obtained with the close BC. We clearly see that, in most of the cases, the upscaled transmissibility computed with the open boundary conditions are greater than the upscaled transmissibility computed with the close boundary conditions. The reason is explained in the images presented in Figure 12. In these pictures are represented only the solutions in the matrix but, due to the high fracture permeability, the pressure field inside the fractures behaves in the same way. Let us consider the two images in the bottom part: in the left we have the solution

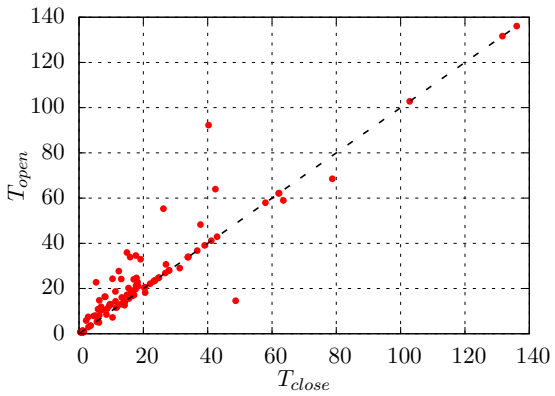

Figure 11: Comparison among the transmissibilities computed using the close (T_{close}) and the open (T_{open}) boundary conditions.

obtained with the close boundary conditions and in the right with the open boundary conditions. We clearly see that the solution obtained with the open BC scales more linearly from the two opposite boundaries of the interface. While the solution obtained with the open BC is more influenced by the presence of the fractures which "propagate" the boundary condition. The pressure across the interface is more flat using the close boundary conditions, thus the flux across the interface is smaller. In this case the different pressure distribution does not compensate this phenomenon and the results are different. In this case we have a transmissibility ration $T_{open}/T_{close} = 2.29$ for the fractures and $T_{open}/T_{close} = 1.34$ for the rock matrix. A similar argument can be used for the two images in the top part of Figure 12. In this case we have a transmissibility ration $T_{open}/T_{close} = 2.84$ for the fractures and $T_{open}/T_{close} = 1.94$ for the rock matrix.

For both approaches, a two-phase flow simulation is performed. The setting and the parameters are the same as in the previous test case. In Figure 13 is depicted the comparison between the two methods with respect to the oil production from the reservoir. In this example, despite of some differences in the upscaled transmissibilities, the productions are almost equal for the entire simulation. Figure 14 represents the water cut for the two methods. In this case some differences can be noticed during the simulation. Nevertheless, at the final time steps, the two values are comparable since almost the 90% of the water is pulled out from the production well.

The selection of different boundary conditions to the local problems may lead to significantly different solutions of the local problems. However, the solution obtained from the two-phase simulator seems to be slightly dependent on the BC applied to the local problems.

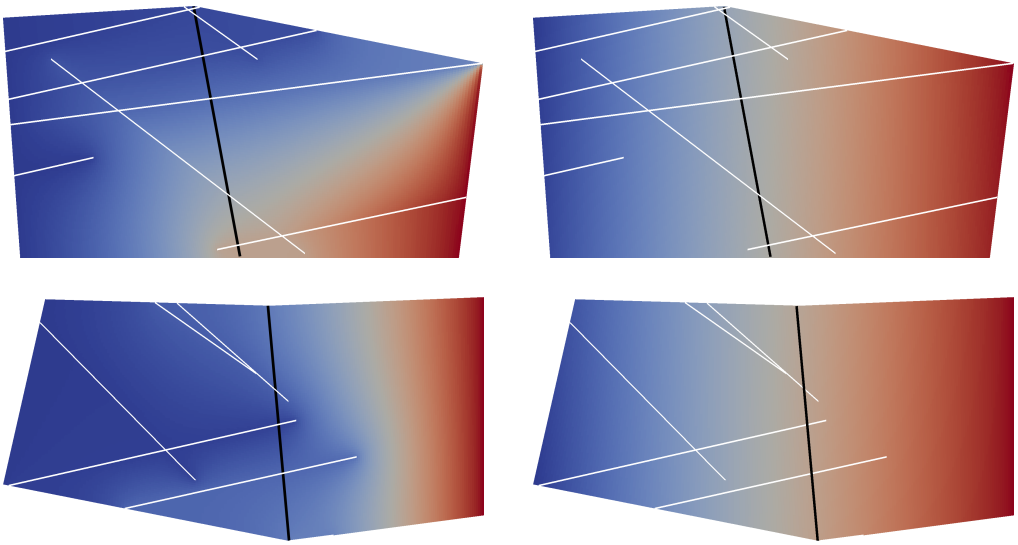

Figure 12: Comparison between the pressure obtained using the close and open boundary conditions. The two images on top are in position $(3,0)-(4,0)$ and the two on bottom are $(4,1)-(5,1)$, according to the (i,j) logic of a corner point grid. The numeration starts from the bottom left corner of the domain.

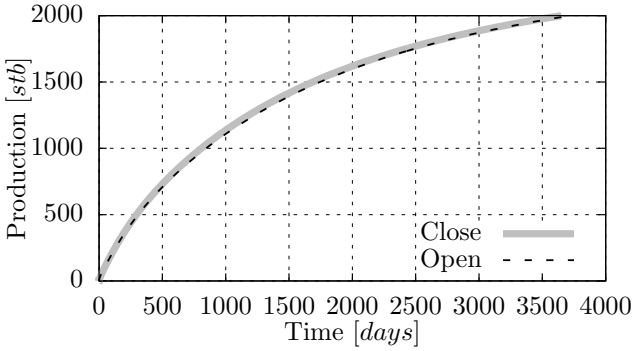

Figure 13: Comparison of the amount of oil produced by the well between the open and close boundary conditions approach.

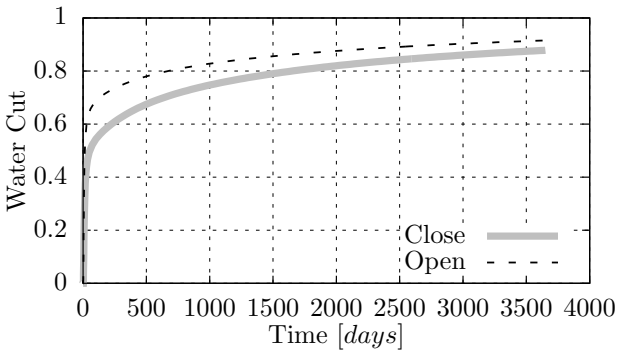

Figure 14: Comparison of the water cut between the open and close boundary conditions approach.

5.3 3D reservoir

In this section we consider a test case, a portion of a bigger realistic reservoir, to make a comparison among the presented approaches. As in the previous test case, the upscaling procedure is applied and the coarse parameters are used to perform a two-phase flow simulation. The computational domain, with the coarse grid and the fractures, is represented in Figure 15. Since the real reservoir is much bigger some of the fractures partly cut the domain whereas others are fully immersed. In the domain are present 1238 fractures organized in 41

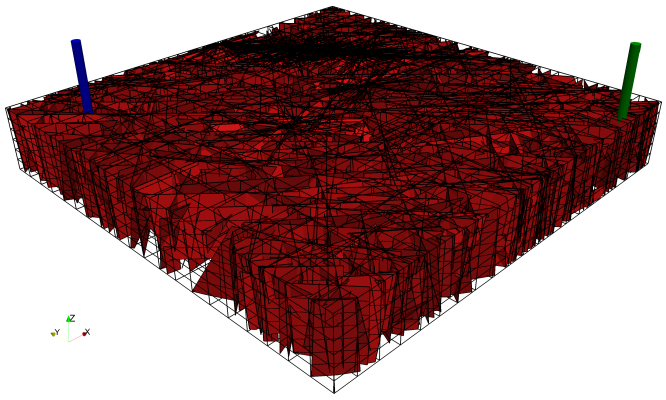

Figure 15: The portion of a real reservoir, with the coarse mesh and fractures, used for the test case in Subsection 5.3. The blue and green cylinder represent the location of the injector and production well, respectively.

networks, but most of them are grouped in a single network composed of 1196 fractures. The coarse grid is made of $20 \times 20 \times 6$ Cartesian-blocks in a domain of sizes $2km \times 2km \times 300m$. The permeability of the fractures has a common value of $6.96 \cdot 10^6 mD$ and of the porous matrix is uniform with value $10^2 mD$. The porosity is 0.876 and 0.16 for the fractures and the porous matrix, respectively.

The fracture thickness ranges from $10^{-4}m$ up to $4 \cdot 10^{-4}m$. For the upscaling procedure, we consider a $10 \times 10 \times 3$ fine grid for each coarse cell. In all the cases two matrix sub-regions are considered from scratch.

We perform four upscaling simulations, by combining the split/no-split option to select whether or not to split the subregions, and the open/close option to select which type of boundary conditions to impose on the problem defined on pairs of cells. The total number of degrees of freedom obtained is equal to 8973 when the splitting algorithm is applied, otherwise we get 7200 degrees of freedom.

Then, for each upscaling simulation, we compute by means of GPRS, a two-phase simulation in order to predict the oil production and water-cut. Initially, the reservoir is saturated by oil (density equal to 49.1 lbm/ft^3). An injector well pushes water (density equal to 64.79 lbm/ft^3) into the reservoir at a constant water flow rate equal to $3 \cdot 10^4$ stb/day. A single production well works at a fixed BHP equal to $100psi$. The location of the injector and production well is sketch in Figure 15, both are completed in the fracture *d.o.f.* of the corresponding coarse cell. The rock compressibility is set equal to $10^{-4}psi^{-1}$. We neglect the residual saturation for both phases and the capillary pressure, and we assume a simple linear relation for the relative permeability. In Figure 16, we plot the oil production for the four considered cases. We notice that the dynamics is slower

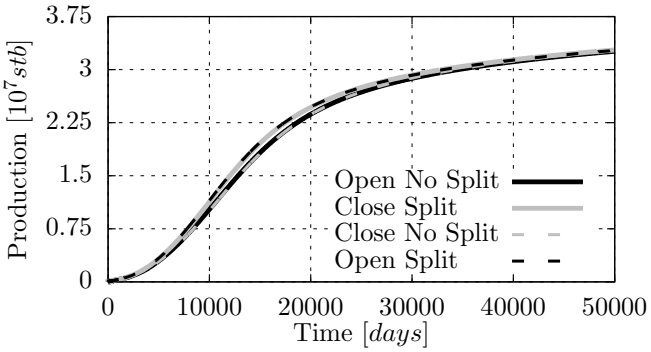

Figure 16: Comparison of the amount of oil produced by the well among the four considered approaches.

than the previous test cases because the reservoir is much bigger as well as the fractures considered are thinner. However, we clearly see that, in this particular case, the solutions are in good agreement, with a slightly over-production if the split algorithm is used. No macroscopic differences are observed considering the open or the close boundary conditions in the oil production. In Figure 17, we plot the water cut dynamics for the four considered cases. As in the oil production all the four solutions are similar, with a more production if the split algorithm is considered, and also in this case with the split algorithm the simulation gives an higher water-cut. No macroscopic differences are observed considering the open or the close boundary conditions in the water-cut.

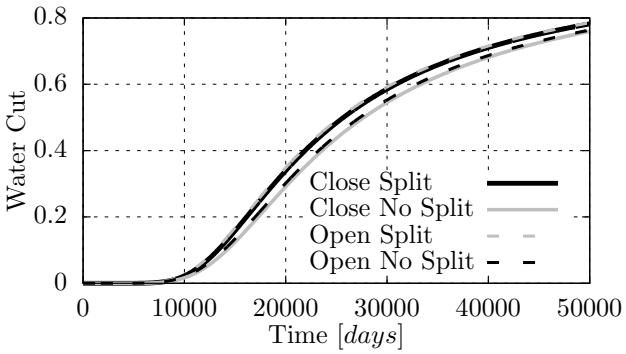

Figure 17: Comparison of the water cut among the four considered approaches.

6 Conclusion

In this work we presented some advances in a flow-based upscaling procedure to compute effective coarse scale parameters of a natural fractured reservoir. Two main improvements were proposed to increase the accuracy keeping the computational cost comparable with the original version of the algorithm. The first one is the splitting of unconnected matrix sub-regions to enhance the local description of the rock matrix. We used a splitting algorithm that can be applied to any kind of computational grids with negligible cost. The other important aspect that we analysed is the numerical comparison of different kinds of boundary conditions that can be imposed in the resolution of the pairs of coarse cells. Both give reasonable and comparable results when applied to synthetic as well as realistic test cases. From what we have observed, if the number of fractures is large enough the macroscopic outcomes, *i.e.* oil production and water-cut, of flow dynamic are similar, whether or not the split algorithm is considered. On the contrary some differences are observed where oil can be trapped in some parts of the rock matrix. A further investigation has to be accomplished in order to examine the case of scarcely connected fracture networks, and the dependence of the upscaled properties with respect to the number of the coarse cells.

One possible improvement is to assign, in each local problem, a different degree of freedom for each unconnected network of fractures. Moreover a parallelization of the proposed method can increase its usability for real scenarios.

7 Acknowledgments

The authors wish to thank Alberto Cominelli, Stefano Micheletti, Paola Panfili, and Anna Scotti for many fruitful discussions. Moreover the authors with to thank Matteo Longoni and Luca Pasquale from Moxoff Srl.

The General Purpose Research Simulator (GPRS) developed by the Reservoir Simulation Research Group (SUPRI-B) at Stanford University was used in this work.

References

[1] Ivar Aavatsmark. Interpretation of a two-point flux stencil for skew parallelogram grids. *Computational Geosciences*, 11(3):199–206, 2007.

[2] Pierre M. Adler, Jean-François Thovert, and Valeri V. Mourzenko. *Fractured porous media*. Oxford University Press, 2012.

[3] Laila Amir, Michel Kern, Vincent Martin, and Jean E. Roberts. Décomposition de domaine et préconditionnement pour un modèle 3D en milieu poreux fracturé. In *Proceeding of JANO 8, 8th conference on Numerical Analysis and Optimization*, December 2005.

[4] Hui Cao. *Development of techniques for general purpose simulators*. PhD thesis, Stanford University, 2002.

[5] Jose Sergio de Araujo Cavalcante Filho, Mahmood Shakiba, Ali Moinfar, and Kamy Sepehrnoori. Implementation of a Preprocessor for Embedded Discrete Fracture Modeling in an IMPEC Compositional Reservoir Simulator. In *SPE Reservoir Simulation Symposium, 23-25 February, Houston, Texas, USA*. Society of Petroleum Engineers, 2015.

[6] Louis J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: a review of recent progress. In *7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany*, pages 23–27, 2003.

[7] Isabelle Faille, Alessio Fumagalli, Jérôme Jaffré, and Jean E. Roberts. Reduced models for flow in porous media containing faults with discretization using hybrid finite volume schemes. Technical report, IFP Energies nouvelles, June 2015. Submitted to: Computational Geosciences.

[8] Alessio Fumagalli, Luca Pasquale, Stefano Zonca, and Stefano Micheletti. An upscaling procedure for fractured reservoirs with non-matching grids. Technical Report 33/2015, Politecnico di Milano, 2015.

[9] Bin Gong, Mohammad Karimi-Fard, and Louis J. Durlofsky. An upscaling procedure for constructing generalized dual-porosity/dual-permeability models from discrete fracture characterizations. In *SPE Annual Technical Conference and Exhibition, San Antonio, Texas*. Society of Petroleum Engineers, 2006.

[10] Yan Gong, Bo Li, and Zhilin Li. Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions. *SIAM J. Numer. Anal.*, 46(1):472–495, 2008.

[11] M.-H. Hui, Bing Gong, Mohammad Karimi-Fard, and Luis J. Durlofsky. Development and Application of New Computational Procedures for Modeling

Miscible Gas Injection in Fractured Reservoirs. In *SPE Annual Technical Conference and Exhibition, 11-14 November, Anaheim, California, U.S.A.* Society of Petroleum Engineers, 2007.

[12] Mohammad Karimi-Fard and Abbas Firoozabadi. Numerical simulation of water injection in fractured media using the discrete-fracture model and the galerkin method. *SPE Reservoir Evaluation & Engineering*, 6(02):117–126, 2003.

[13] Mohammad Karimi-Fard, Bin Gong, and Luis J. Durlofsky. Generation of coarse-scale continuum flow models from detailed fracture characterizations. *Water Resources Research*, 42(10), 2006.

[14] Liyong Li and Seong H. Lee. Efficient field-scale simulation of black oil in a naturally fractured reservoir through discrete fracture networks and homogenized media. *SPE Reservoir Evaluation & Engineering*, 11:750–758, 2008.

[15] Vincent Martin, Jérôme Jaffré, and Jean E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691, 2005.

[16] Ali Moinfar, Abdoljalil Varavei, Kamy Sepehrnoori, and Russell T. Johns. Development of an efficient embedded discrete fracture model for 3d compositional reservoir simulation in fractured reservoirs. *Society of Petroleum Engineers*, 19(2):289–303, April 2014.

[17] Paola Panfili and Alberto Cominelli. Simulation of miscible gas injection in a fractured carbonate reservoir using an embedded discrete fracture model. In *Abu Dhabi International Petroleum Exhibition and Conference, 10-13 November, Abu Dhabi, UAE.* Society of Petroleum Engineers, 2014.

[18] Paola Panfili, Alberto Cominelli, and Anna Scotti. Using Embedded Discrete Fracture Models (EDFMs) to Simulate Realistic Fluid Flow Problems. In *Second EAGE Workshop on Naturally Fractured Reservoirs, Muscat, Oman*, 2013.

[19] A. B. Tatomir, A. Szymkiewicz, H. Class, and R. Helmig. Modeling two phase flow in large scale fractured porous media with an extended multiple interacting continua method. *Computer Modeling in Engineering & Sciences*, 77(2):81–112, 2011.

[20] Xiao-Hui Wu, Matthew Stone, Rossen Parashkevov, David Stern, and Stephen Lincoln Lyons. Reservoir modeling with global scale-up. In *SPE Middle East Oil and Gas Show and Conference, 11–14 March, Manama, Bahrain.* Society of Petroleum Engineers, 2007.