A multiscale finite element method for simulating flow in fractured porous media

Nathan Shauer[1], Pedro Lima[1], Jose B. Villegas S.[2], Philippe R. B. Devloo[1]

[1]*LabMeC-FECAU-Universidade Estadual de Campinas*
R. Josiah Willard Gibbs 85, Cidade Universitária, 13083-841, Campinas-SP, Brazil
shauer@unicamp.br, pedrolimasi@outlook.com, phil@unicamp.br
[2]*Faculdade de Ciencias de la Ingeniería-Universidad Estatal Peninsula de Santa Elena*
Km 1 vía La Libertad - Salinas, 240204, Santa Helena, Ecuador
jvillegas@upse.edu.ec

Abstract. This work extends on Durán et al. [1] to propose a multiscale locally conservative finite element method for the simulation of flow in fractured porous media. The method employs H(div)-confirming flux approximations that carry advantages such as the ability to solve problems with nearly incompressible materials, better accuracy for the velocity field approximation, fewer requirements on the regularity of the solution, and continuity of the normal velocity between elements. The last of these leads to locally conservative approximations of the velocity field, which is considered paramount in the area of reservoir simulation. The flow in the porous media is modeled using traditional Darcy's law, and the coupling with the fracture flow is modeled with the Discrete-Fracture-Matrix representation, where the fractures are idealized as lower-dimensional elements at the interface of matrix elements. The method is applied to a benchmark problem of a complex reservoir with several fractures.

Keywords: Discrete Fracture Networks, Porous Media Flow, Multiscale Method, Mixed Finite Elements

1 Introduction

More than half of the oil reservoirs in the world are fractured (Schlumberger [2]). These fractures are paths of high conductivity that can strongly affect the flux pattern in the porous media. Consequently, the recent decade has observed increasing trends of research into flow in fractured porous media (see reviews by [3, 4]). This is primarily motivated by subsurface processes as in the case of fractured porous media but also stems from research in other fields such as materials science and biological applications [5]. Therefore, being able to accurately simulate fluid flow in fractured porous media can be of great value to the field.

This article presents a multiscale, locally conservative methodology to simulate flow in fractured porous media. The flow in the porous media is assumed to be governed by traditional Darcy's law and the coupling with the fractures is based on the Discrete-Fracture-Matrix method, where the fractures are represented by two-dimensional entities at the interface of bulk matrix three-dimensional entities. A form of Darcy's law is adopted for the flux inside the fractures where the permeability is based on the fracture opening. The governing equations are solved using the Mixed Finite Element Method where the fluxes are approximated using H(div) approximation spaces. This method leads to locally conservative approximations with optimal flux accuracy. Additionally, the Multiscale Hybrid-Mixed Method (MHM) is used where the problem domain is divided into macro domains and the global system of equations is mostly composed of degrees of freedom related to the normal flux between these macro domains. Lastly, a mesh generator tailored for generating multiscale DFM meshes is presented.

2 Methodology

In this section, the governing equations for fluid flow in fractured porous media are presented. The FEM adopted to discretize these equations and a new methodology to generate meshes that are tailored for this problem are also briefly presented. More details can be found in Durán et al. [1], Berre et al. [4] and Lima [6].

2.1 Strong form of the governing equations

The discrete-fracture-matrix model considers flow both in the fractures and in the surrounding porous media. With a steady-state, incompressible, single-phase flow described by Darcy's law, the boundary value problem for flow in the porous matrix can be stated as:

Find the fluid velocity $\mathbf{u}_3$ and the hydraulic head p_3 that is the solution of the following system of Partial Differential Equations:

$$\begin{cases} \mathbf{K}_3^{-1}\mathbf{u}_3 + \nabla p_3 & = & 0 & \text{in } \Omega_3 \\ \nabla \cdot \mathbf{u}_3 & = & 0 & \text{in } \Omega_3 \\ p_3 & = & p_{3_D} & \text{on } \partial\Omega_{3_D} \\ \mathbf{u}_3 \cdot \mathbf{n}_3 & = & u_{3_N} & \text{on } \partial\Omega_{3_N} \end{cases}, \tag{1}$$

where $\mathbf{K}_3$ is the hydraulic conductivity, Ω_3 is the reservoir domain, $\partial\Omega_{3_D}$ is the part of the domain boundary with imposed hydraulic head p_{3_D}, $\partial\Omega_{3_N}$ is the part of the domain boundary with imposed normal velocity u_{3_N}, and $\mathbf{n}_3$ is the outward normal on $\partial\Omega_{3_N}$.

Similarly, the following governing equations hold for fluid flow inside the fracture:

$$\begin{cases} \frac{1}{a_2}\mathbf{K}_2^{-1}\mathbf{u}_2 + \nabla p_2 & = & 0 & \text{in } \Omega_2 \\ \nabla \cdot \mathbf{u}_2 - \sum \mathbf{u}_3 \cdot \mathbf{n}_{\Gamma_2} & = & 0 & \text{in } \Omega_2 \\ p_2 & = & p_{2_D} & \text{on } \partial\Omega_{2_D} \\ \mathbf{u}_2 \cdot \mathbf{n}_2 & = & u_{2_N} & \text{on } \partial\Omega_{2_N} \end{cases}, \tag{2}$$

where $\mathbf{K}_2$ is the fracture pseudo-permeability, a_2 is the fracture opening, $\mathbf{u}_2$ is the fluid velocity inside the fracture, p_2 is the fluid hydraulic head inside the fracture, Ω_2 is the fracture cavity domain, $\partial\Omega_{2_D}$ is the part of the fracture domain boundary with imposed hydraulic head p_{2_D}, $\partial\Omega_{2_N}$ is the part of the fracture domain boundary with imposed normal velocity u_{2_N}, $\mathbf{n}_2$ is the outward normal on $\partial\Omega_{2_N}$, and $\mathbf{n}_{\Gamma_2}$ is the outward normal on the fracture faces.

Additionally, at the interface between the reservoir matrix and fracture faces the pressure gradient can be approximated as $(p_2 - p_3)/a_2$. Then, the following relation holds at the interfaces (Berre et al. [4]):

$$\mathrm{u}_3 \cdot \mathbf{n} + 2\frac{K_2^{eq}}{a_2}(p_2 - p_3) = 0 \quad \text{on } \Gamma_2, \tag{3}$$

where K_2^{eq} is the fracture equivalent tangential permeability, and Γ_2 is the fracture surface (faces).

2.2 Mixed Finite Elements discretization

Equations eq. (1) and eq. (2) are approximated using the Mixed Finite Element Method (MFEM) described in Castro et al. [7]. The MFEM is a type of Finite Element technique where both the state variable and flux fields are solved as unknowns of the problem. In the case of porous media flow, this translates to the hydraulic head (state variable) and the velocity field (flux). Compared to the traditional Finite Element Method (FEM), the MFEM carries advantages such as the ability to solve problems with nearly incompressible materials, better accuracy for the velocity field approximation, less requirements on the regularity of the solution, and continuity of the normal velocity between elements. The last of these leads to locally conservative approximations of the velocity field, which is considered paramount in the area of reservoir simulation. Details on the adopted MFEM can be found in Durán et al. [1], Castro et al. [7], Devloo et al. [8].

2.3 The Multiscale Hybrid-Mixed Method

The Multiscale Hybrid-Mixed Method (MHM) is a methodology tailored to tackle problems that have strongly varying material properties. It uses a divide-and-conquer strategy to partition the mesh into so-called macro domains. For each macro domain, different discretization techniques can be adopted. The contributions of each macro domain are then condensed into an approximation space defined in the interface between macro domains, leading to a global system of equations with reduced size. A noteworthy feature of the method is that each macro domain is independent and, therefore, can be solved in parallel in a multi-core computer. An illustration of an MHM mesh with four macro domains is shown in Figure Fig. 1. More details on the MHM can be found in Durán et al. [1].

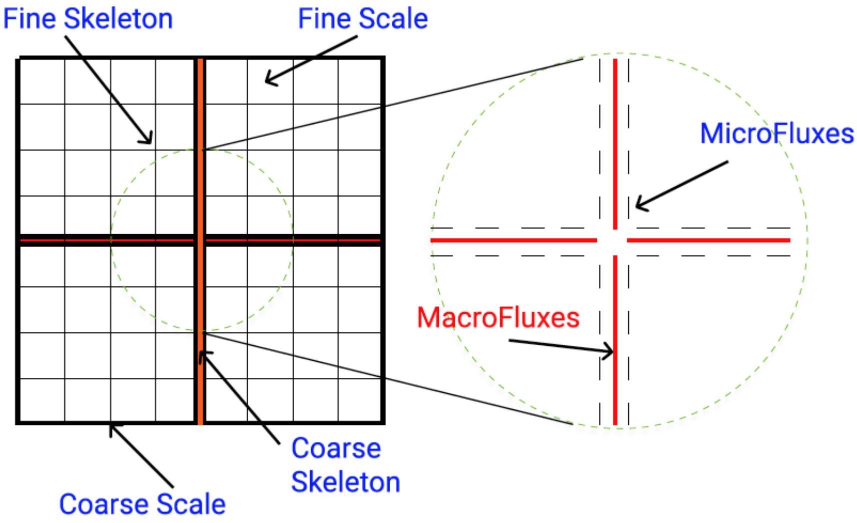


Figure 1. Example of MHM coarse and fine meshes. The coarse mesh is composed of 4 quadrilateral elements and each macro element has a 4x4 uniform refinement leading to 16 quadrilateral microelements. The global system is only composed of normal macro fluxes at the interfaces and constant pressure at each macro element.

2.4 DFNMesh: A Discrete-Fracture-Matrix mesh generator

In this work, a mesh generator that is tailored for DFM problems discretized with multiscale methodologies is used (Lima et al. [9]). This mesh generator is based on the iterative refinement of an initial coarse mesh composed of convex polyhedra to obtain a mesh that conforms with the location of the fractures in the domain. The method can handle the complete multi-dimensional domain from the tridimensional porous rock matrix, through fracture surfaces, down to open curves for fracture-fracture intersections. The main steps involve: intersect edges by checking for nodes on opposite sides of the fracture plane, extend intersections from edges to faces, coalesce intersections to closest existing nodes (given a tolerance), refine interface elements to conform to the fracture, identify subsets of fracture surface, mesh the surface, and locate boundaries and intersections where they arise. Finally, the space around fractures is filled with the fine-scale unstructured mesh, which is kept conformal. The coalescence of intersections to closest existing nodes is performed based on two criteria:

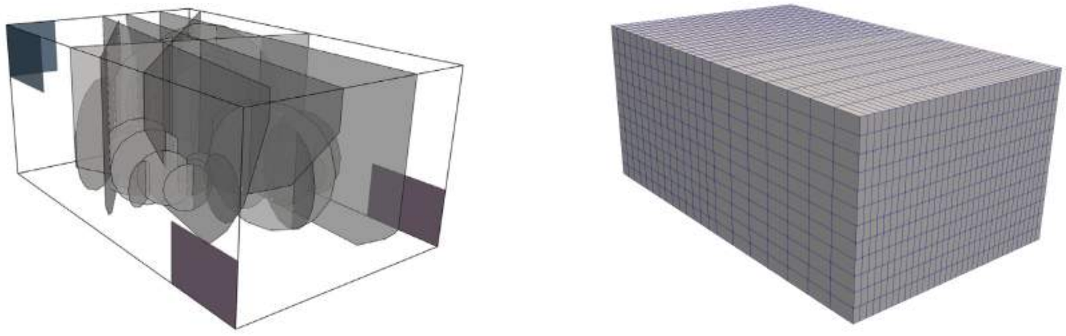
1. **Points closer than a tolerance to existing points are rejected/snapped**: Intersection points are only created in intersected edges. Hence, this tolerance check is simply a measurement of the distance of the intersection point to both nodes of the 1D side. If any distance is smaller than the tolerance, the point is discarded for the closest node. This tolerance is referred to as ϵ_d.
2. **Angle below a tolerance**: Within the domain of the skeleton mesh, the angles an intersecting fracture creates (as it refines necessary elements) are the internal angles of the sub-elements. All these angles are checked against a tolerance. To impose this tolerance, note that all sub-elements have at least one corner defined by one of the intersection points the fracture plane has created at an edge of its father. If any internal angle violates the tolerable angle, it is enforced by coalescing the intersection point to a corner of the father element. This tolerance is referred to as ϵ_α.

The entirety of the DFNMesh code is written in `C++` and hosted open-source at `https://github.com/labmec/DFNMesh`. It largely relies on two other open-source finite element libraries: NeoPZ ([10]) and Gmsh ([11]).

3 Example

3.1 Benchmark problem with 52 fractures

This test is designed to test a DFM simulator's robustness in the presence of a large number of fractures. The problem domain is defined on $\Omega = [-500\,\mathrm{m}, 350\,\mathrm{m}] \times [100\,\mathrm{m}, 1500\,\mathrm{m}] \times [-100\,\mathrm{m}, 500\,\mathrm{m}]$ and 52 fractures are present as illustrated in Figure 2.



(a) Problem domain, part of the boundary with imposed flow (blue), and part of the boundary with imposed pressure (purple). This figure is adopted from [4].

(b) Initial coarse mesh adopted for the DFNMesh algorithm.

Figure 2. Illustration of problem domain and initial coarse mesh for Benchmark Problem.

A constant entering flux of $u_{2_N} = -1.0\,\mathrm{m/s}$ is imposed at $\partial\Omega_{2_{N,1}} = \{(x, y, z) \in \partial\Omega : (-500\,\mathrm{m}, -200\,\mathrm{m}) \times \{1500\,\mathrm{m}\} \times (300\,\mathrm{m}, 500\,\mathrm{m})\}$ and $\partial\Omega_{2_{N,2}} = \{(x, y, z) \in \partial\Omega : \{-500\,\mathrm{m}\} \times (1200\,\mathrm{m}, 1500\,\mathrm{m}) \times (300\,\mathrm{m}, 500\,\mathrm{m})\}$, and a condition of $p_3 = 0\,\mathrm{m}$ is imposed at $\partial\Omega_{2_{D,1}} = \{(x, y, z) \in \partial\Omega : \{-500\,\mathrm{m}\} \times (100\,\mathrm{m}, 400\,\mathrm{m}) \times (-100\,\mathrm{m}, 100\,\mathrm{m})\}$ and $\partial\Omega_{2_{D,2}} = \{(x, y, z) \in \partial\Omega : \{350\,\mathrm{m}\} \times (100\,\mathrm{m}, 400\,\mathrm{m}) \times (-100\,\mathrm{m}, 100\,\mathrm{m})\}$. Zero flux is applied to the remaining part of the boundary of the 3D reservoir domain. The adopted material properties are $\mathbf{K}_3 = \mathbf{I}\,\mathrm{m/s}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{K}_2 = 100\,\mathbf{I}\,\mathrm{m}^3/s$, $K_2^{eq} = 10^4$ m/s, and $a_2 = 10^{-2}$ m.

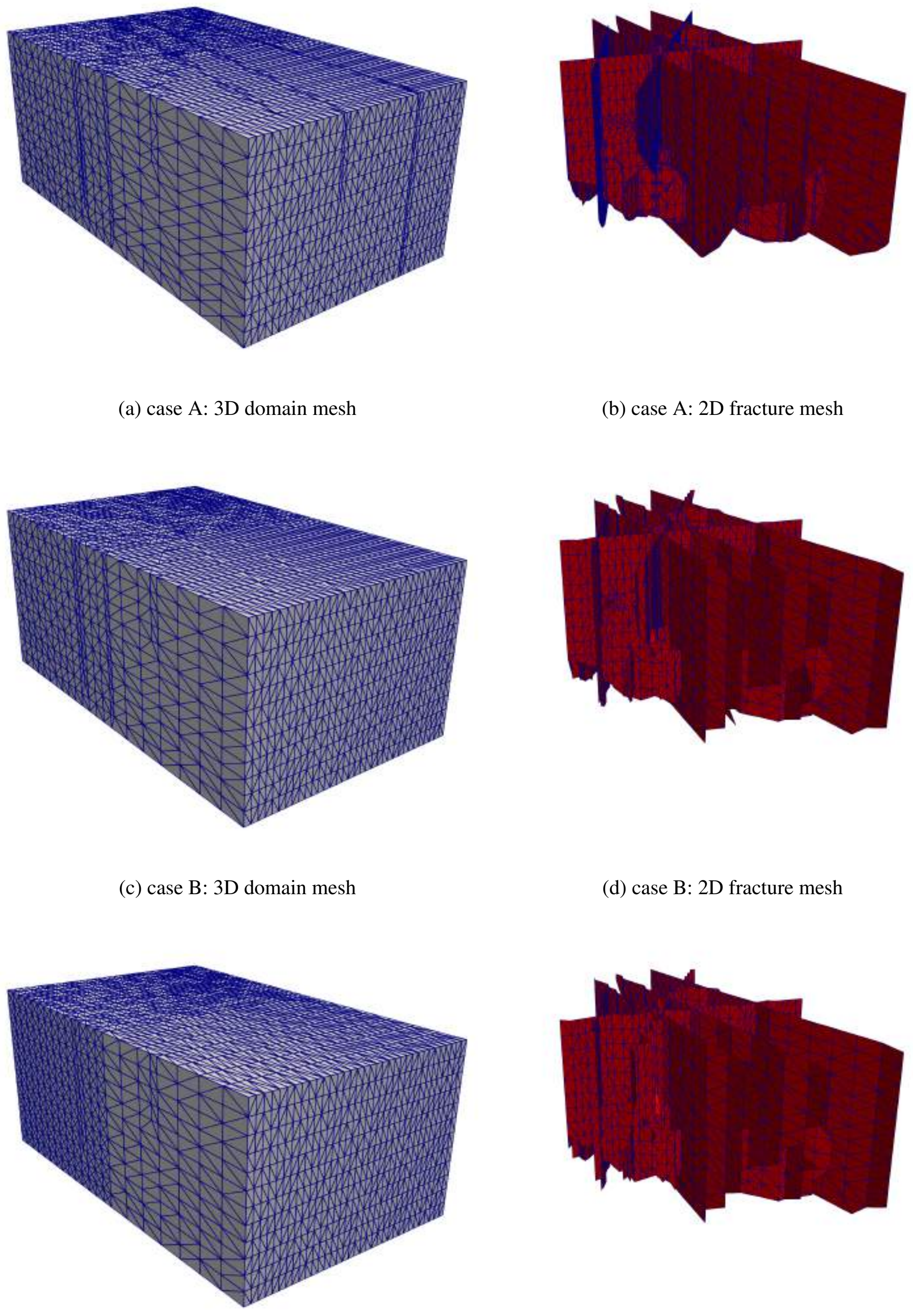
Mesh generation

Three different meshes were generated using the DFNMesh algorithm applied to the initial coarse mesh shown in Figure 2 and different snapping tolerances. It is noted that the initial mesh is more refined on the back due to high concentration of fractures in that region. The cases are denoted A, B, and C with the tolerances adopted shown in Table 1. The generated meshes are shown in Figure 3. What can be clearly seen is that case A most accurately represents the problem geometry, and cases B and C progressively less accurately represent the geometry of the fractures.

The gradually stronger imposition of geometrical tolerances from cases A to C induces an improvement in mesh quality. Figure 4 collects gamma quality distribution in histograms for each of the tolerance cases, and shows the trend of mesh improvement as feature rejection gets more aggressive.

Table 1. Snapping tolerances adopted for each case considered in the analysis of Benchmark Problem.

Case	ϵ_d	ϵ_α
A	1×10^{-1}	1.5×10^{-2}
B	4.5×10^{-1}	2.75×10^{-1}
C	5×10^{-1}	3×10^{-1}



(a) case A: 3D domain mesh

(b) case A: 2D fracture mesh

(c) case B: 3D domain mesh

(d) case B: 2D fracture mesh

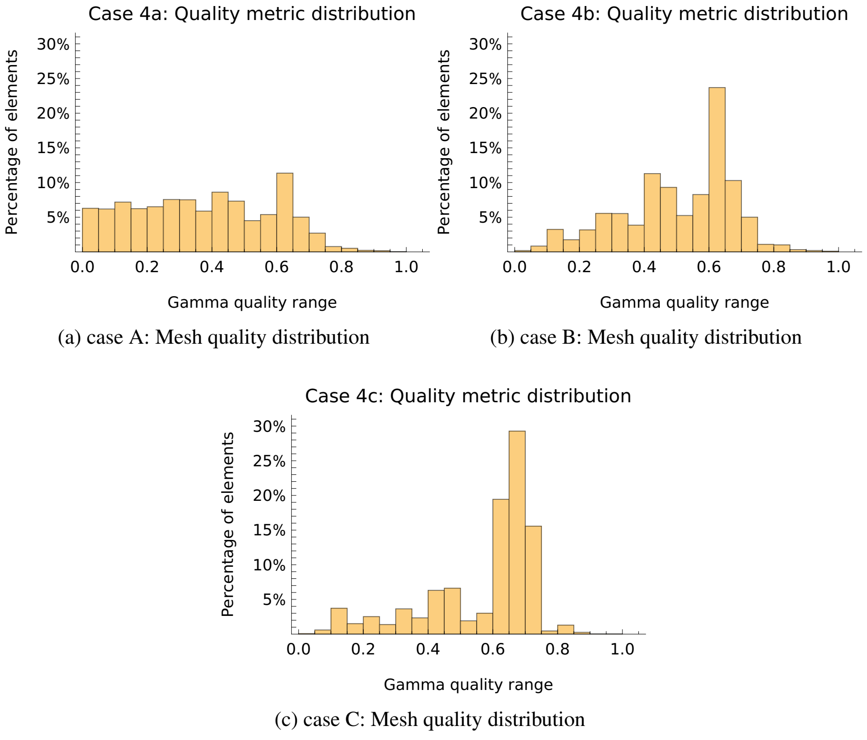
(e) case C: 3D domain mesh

(f) case C: 2D fracture mesh

Figure 3. Finite element meshes used in the analysis of benchmark problem. The 3D domain meshes are shown in the left column and the 2D meshes representing the fractures are shown in the right column. Four cases are considered where different snapping tolerances are adopted. Case A adopts tolerances that lead to the most accurate representation of the problem geometry and case C adopts tolerances that lead to the less accurate representation of the problem geometry.

Results

In [4], a plot of the hydraulic head along a line passing through $(350\,\mathrm{m}, 1500\,\mathrm{m}, 500\,\mathrm{m})$ and $(-500\,\mathrm{m}, 100\,\mathrm{m}, -100\,\mathrm{m})$ is chosen as the parameter for comparison of different discretization techniques. In Figure 5, the results obtained in [4] are compared with the results of the MFEM adopted in this work in conjunction with the three meshes



(a) case A: Mesh quality distribution

(b) case B: Mesh quality distribution

(c) case C: Mesh quality distribution

Figure 4. Histograms for distribution of element gamma quality in fine meshes of benchmark problem created for each case of geometrical tolerances. Elements are grouped by ranges of 0.05 gamma values and presented in percentage. Three cases are considered where different snapping tolerances are adopted. Case A adopts tolerances that lead to the most faithful representation of the problem geometry and less snapping of features. Case C coalesces a higher number of features, which improves gamma quality but distorts the mesh away from user input.

presented in Section 3.1. The results obtained by the MFEM using the meshes of cases A, B, and C are within the range of the benchmark results by other methods, demonstrating that the meshes generated by the DFNMesh algorithm can generate consistent results regardless of the snapping tolerances chosen even for a problem with a large number of fractures.

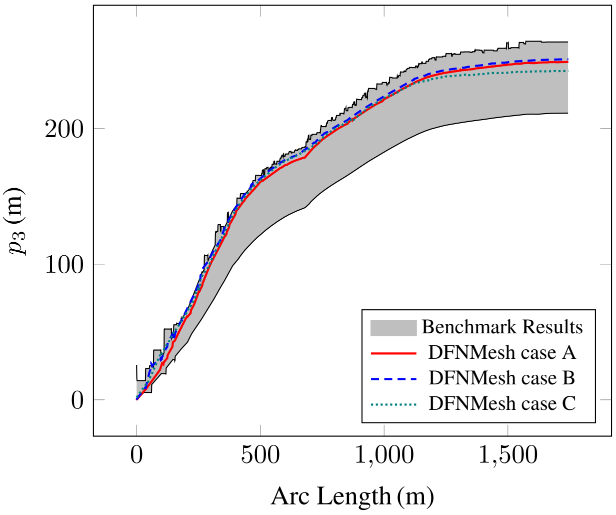


Figure 5. Comparison of matrix hydraulic head over line through domain between the four meshes adopted using the MFEM and other methods available for Benchmark Problem (Berre et al. [4]).

4 Conclusions

A multiscale, locally conservative methodology to simulate flow in fractured porous media is presented where the global degrees of freedom are mostly due to normal fluxes between macro domains. The methodology has the computational advantage that each macro domain can be solved independently, leading to a high degree of parallelization. A mesh generator tailored to generate meshes for this multiscale method is also shown which is open-source and can be readily downloaded from Github. This mesh generator lets the user decide how much degree of geometric fidelity they require. Nevertheless, the obtained results, even for less geometrically accurate meshes, are well within the acceptable range. These less geometrically accurate meshes have the benefit of possessing more well-behaved elements with better aspect ratios that can decrease numerical instability issues caused by deteriorated conditioning of the stiffness matrices.

Acknowledgements. We gratefully acknowledge the support of EPIC - Energy Production Innovation Center, hosted by the University of Campinas (Unicamp) and sponsored by Equinor Brazil and FAPESP - São Paulo Research Foundation (Process 2017/15736-3, N. Shauer FAPESP scholarship 2021/03791-5. We acknowledge the support of ANP (Brazilian National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation. Acknowledgements are extended to the Center for Petroleum Studies (Cepetro), and the Faculty of Civil Engineering, Architecture and Urbanism (FECFAU) of Unicamp. The author P. R. B. Devloo is grateful for the financial support from the Brazilian Research Council (CNPq), grant 305823/2017-5.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] O. Durán, P. R. Devloo, S. M. Gomes, and F. Valentin. A multiscale hybrid method for darcy's problems using mixed finite element local solvers. *Computer Methods in Applied Mechanics and Engineering*, vol. 354, pp. 213–244, 2019.

[2] Schlumberger. Carbonate reservoirs: Meeting unique challenges to maximize recovery. Technical Report, Schlumberger, Houston, 2007.

[3] N. Schwenck, B. Flemisch, R. Helmig, and B. I. Wohlmuth. Dimensionally reduced flow models in fractured porous media: crossings and boundaries. *Computational Geosciences*, vol. 19, n. 6, pp. 1219–1230, 2015.

[4] I. Berre, W. M. Boon, B. Flemisch, A. Fumagalli, D. Gläser, E. Keilegavlen, A. Scotti, I. Stefansson, A. Tatomir, K. Brenner, S. Burbulla, P. Devloo, O. Duran, M. Favino, J. Hennicker, I.-H. Lee, K. Lipnikov, R. Masson, K. Mosthaf, M. G. C. Nestola, C.-F. Ni, K. Nikitin, P. Schädle, D. Svyatskiy, R. Yanbarisov, and P. Zulian. Verification benchmarks for single-phase flow in three-dimensional fractured porous media. *Advances in Water Resources*, vol. 147, pp. 103759, 2021.

[5] K. Erbertseder, J. Reichold, B. Flemisch, P. Jenny, and R. Helmig. A coupled discrete/continuum model for describing cancer-therapeutic transport in the lung. *PLoS ONE*, vol. 7, n. 3, 2012.

[6] P. Lima. Automatic multiscale meshing for three-dimensional discrete fracture networks. Master's thesis, Universidade Estadual de Campinas, 2021.

[7] D. A. Castro, P. R. Devloo, A. M. Farias, S. M. Gomes, D. de Siqueira, and O. Durán. Three dimensional hierarchical mixed finite element approximations with enhanced primal variable accuracy. *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 479–502, 2016.

[8] P. R. B. Devloo, W. Teng, and C. S. Zhang. Multiscale hybrid-mixed finite element method for flow simulation in fractured porous media. *CMES - Computer Modeling in Engineering and Sciences*, vol. 119, n. 1, pp. 145–163, 2019.

[9] P. Lima, N. Shauer, J. B. Villegas, and P. R. Devloo. Dfnmesh: Finite element meshing for discrete fracture matrix models. *Manuscript submitted*, 2023.

[10] P. R. B. Devloo. PZ: An object oriented environment for scientific programming. *Computer Methods in Applied Mechanics and Engineering*, vol. 150, n. 1-4, pp. 133–153, 1997.

[11] C. Geuzaine and J. F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, vol. 79, n. 11, pp. 1309–1331, 2009.