MS-DFM

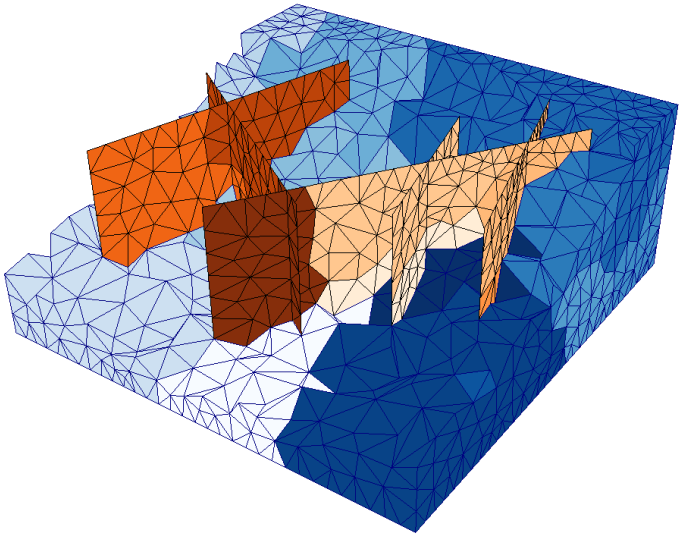
A Multiscale Finite Volume Method for Discrete Fracture Modeling on Unstructured Grids

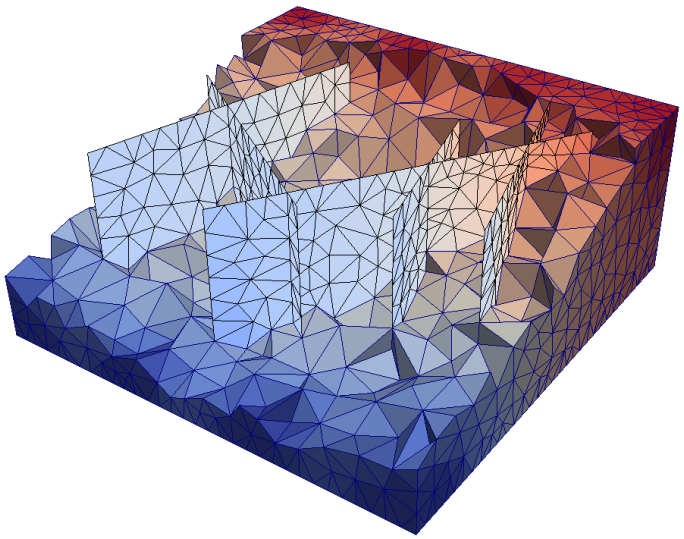
S.B.M. Bosma

MSc Thesis

Faculty of Civil Engineering & Geosciences, TU Delft

16 February 2017





TU Delft

MS-DFM

A Multiscale Finite Volume Method for Discrete Fracture Modeling on Unstructured Grids

by

S.B.M. Bosma

in partial fulfillment of the requirements for the degree of Master of Science in Applied Earth Sciences
at the Delft University of Technology,
to be defended publicly on Thursday February 16, 2017 at 14:30.

Student number:	4077504	
Thesis committee:	Dr. H. Hajibeygi,	Geoscience and Engineering, TU Delft; principal supervisor
	Prof. dr. ir. J.D. Jansen,	Geoscience and Engineering, TU Delft
	Dr. D.V. Voskov,	Geoscience and Engineering, TU Delft
	Prof. dr. ir. M. Bakker,	Watermanagement, TU Delft
Advisors:	Prof. H. Tchelepi,	Energy Resources Engineering, Stanford University; co-supervisor
	Ir. M. Tene,	Geoscience and Engineering, TU Delft;

This thesis is confidential and cannot be made public until February 13, 2017.

TU Delft

Abstract

A novel multiscale method for discrete fracture modeling on unstructured grids (MS-DFM) is developed. To this end, the DFM fine-scale discrete system is constructed using unstructured conforming cells for the matrix with lower-dimensional fracture elements placed at their interfaces. On this unstructured fine grid, MS-DFM imposes independent unstructured coarse grids for the fracture and matrix domains. While the conservative coarse-scale system is solved over these coarse-grid cells, overlapping dual-coarse blocks are also formed in order to provide local supports for the multiscale basis functions. To increase the accuracy, but maintaining the computational efficiency, fracture-matrix coupling is considered only for the basis functions inside the matrix domain. This results in additional (enriching) fracture basis functions in the matrix. By construction, basis functions form the partition of unity for both fracture and matrix sub-domains. Furthermore, to enable error reduction to any desired level, a convergent iterative strategy is developed, where MS-DFM is employed along with a fine-scale smoother in order to resolve low- and high-frequency modes in the error. The performance of MS-DFM is assessed for several 2D and 3D test cases. Specially, the method proves accurate for many test cases without any iterations. MS-DFM is the first of its kind, and thus extends the application of multiscale methods to unstructured discrete fracture models. As such, it provides a promising framework for real-field application of unstructured DFM.

Preface

Before you lies the final product of over a year of research. It marks the end of my Master's degree and the culmination of my academic career up to now. The project has been a journey which has sparked my passion for research and has granted me opportunities which I have long dreamt of. The endeavor has been as much a team effort as it has been a personal one, and I am fortunate to have had such world-class support and company along the way.

Walking into Dr. Hajibeygi's office 2 years ago, I didn't really know what I was getting myself into. I knew I wanted to experience academic research and pursue a quantitative topic. Looking back now, I can say the decision was a pure homerun. Hadi's infectious passion and enthusiasm is unrivaled and has been of invaluable contribution to this work. From white-boards filled with equations to devising algorithms in the library courtyard, thank you for all your knowledge and support throughout this project. I have thoroughly enjoyed working together and hope our career paths cross again in the future.

Another important influence of this work is Prof. Tchelepi. His foresight early on in the code development in addition to his goal-driven approach led to a smooth and efficient process. Moreover, I would like to thank him for the opportunity to visit the Supri-B team at Stanford and his encouraging supervision which gave me freedom to try out new ideas while still being guided in the right direction.

The work would not have been possible without the support of Matei Tene. His help with learning c++ and code debugging allowed me and the research to excel rapidly. Additionally, the numerous discussions and presentations with the DARSIM group at the TU Delft allowed me to access a broader pool of knowledge. My thanks to the whole team.

I would like to thank prof. Jansen not only for being a part of my committee but also for his important support for my future career. Furthermore, thank you Dr. Voskov for your valuable feedback during presentations which helped put my work into perspective and encouraged me to think one step further. I would also like to thank Prof. Bakker for assessing the work and being a part of my committee.

Last but not least, I owe a big thank you to my friends and family for their endless support and joy-filled encouragement. Martijn, Marco and Patricia, thank you for putting up with my thoughts about triangles and pyramids while on vacation. Yvonne, Arnaud, Andy, and my housemates at OD122, thank you for all the fun and for energizing me on brain-fried evenings after days of programming.

Without further ado, I am proud to present to you the result of our work and I hope you enjoy the read!

S.B.M. Bosma
Delft, February 2017

Contents

List of Figures

Nomenclature

This list describes several symbols that will be later used within the body of the document.

$\breve{\bullet}$	Coarse scale value
λ	Mobility
$\mathbf{bf_i}$	Set of basis functions contributing to cell i
$\mathbf{k}_i$	Permeability of cell i
$\mathbf{P}$	Prolongation Operator
$\mathbf{R}$	Restriction Operator
Φ	Basis function
$\vec{n}_f$	Normal vector to the face-plane
$\vec{n}_f$	Unit normal vector to the face
$\vec{v}$	Vector from the grid node projection to the grid node itself
$\vec{v}_{cf}$	Vector from cell center to face
A_f	Face area
N_{CF}	Number of coarse fracture unknowns
N_{CM}	Number of coarse matrix unknowns
p	Pressure
p'	Pressure approximation
q	Source terms
T_{if}	Half-transmissibility from the center of cell i to interface f
T_{ij}	Transmissibility from cell i to cell j

1 Introduction

Fractures play a crucial role in flow and transport of mass and heat in many subsurface engineering applications, including (conventional and unconventional) hydrocarbon and enhanced geothermal reservoirs [21]. As lower-dimensional features with multiple length scales and highly contrasting flow properties, they impose a significant challenge to numerical simulation of subsurface flow. As such, development of accurate and efficient methods for fractured porous media is essential for the next-generation simulators.

A widely studied approach in the reservoir simulation community is discrete fracture modeling, where fractures are modeled as lower dimensional features within the matrix rock. To allow for independent matrix and fracture grids, the embedded discrete fracture modeling (EDFM) approach has been developed [6, 12, 13]. More recently, the projection-based EDFM method enables the simulation of a wider range of conductivity contrasts including flow barriers [26, 27]. Alternatively, a different approach uses unstructured grids to place fractures at the interface between matrix cells [1, 9, 20]. By doing so, unstructured DFM, or more commonly DFM, accurately represents geological input data, including fracture geometry, and allows for a convenient and accurate discretization scheme. For example, impermeable fractures as well as matrix-fracture interactions do not require special treatment. Furthermore, note that both methods, EDFM and DFM, are compatible with hierarchical fracture modeling where small fractures are upscaled into matrix cell properties [19]. The flexibility and accuracy of DFM comes at a cost: for real field application, DFM leads to large linear systems that are beyond the scope of classical solution strategies. Especially for fractured media, traditional solutions such as excessive upscaling of flow-related quantities can lead to inaccurate solutions. To resolve this challenge, in this work, a multiscale DFM method is proposed.

Multi-Scale (MS) methods have been developed to reduce the computational complexity of flow simulation in highly heterogeneous media [3, 7, 8]. The MS strategy consists of solving local fine scale problems and a corresponding coarse scale system to accurately approximate the considerably-sized original fine scale problem. To do so, these methods first divide the global domain into small overlapping problems which are then solved to obtain basis functions (or interpolators). Next, a coarse system is constructed in which the coarse cell transmissibility values are computed using these basis functions. The obtained coarse solution is interpolated back to the fine-scale resolution using the same basis functions. This way, MS methods provide approximate solutions at the original problem resolution, while preserving the fine-scale information. Such a procedure naturally allows for systematic error-reduction strategies, namely, iterative multiscale methods [5, 25, 30, 33]. Extending MS methods for fracture simulation, recent works include the flexible coupling of matrix-fracture basis functions with the EDFM approach [16]. To allow for reconstruction of the mass conservative fine-scale velocity from the approximate multiscale solution, note that a finite-volume coarse-scale system is required [2, 4, 8, 11, 34].

Despite multiple extensions of MS methods, the majority have been applied to structured Cartesian grids. Additionally, the limited investigations into unstructured grids have either restricted coarse grid flexibility or kept the third dimension structured (2.5D) [15, 17, 22]. With the exception of the latter reference, where an MS-EDFM approach was developed on 2.5D domains, the challenge of combining unstructured grids and MS methods for 3D fractured porous media has not yet been tackled nor resolved.

In this work, a multiscale method for the Discrete Fracture Modeling approach on fully unstructured 3D grids (MS-DFM) is devised. A similar approach to the MS-EDFM method [16] is followed, where we extend the state-of-the-art MS methods to include complex grids within the DFM formulation. Starting from 2 independent arbitrary partitions, i.e., one for the matrix and one for the fractures, a 3D wire-basket ordering [29, 32] is constructed based on the dual-coarse grid in each domain. The matrix and fracture dual-grids have 4 (internal, face, edge, node) and 3 (face, edge, node) levels of hierarchy, respectively.

The aforementioned basis functions are first solved in the fracture domain, with local supports provided by the fracture dual grid and no coupling towards the matrix. The two media are coupled through the basis functions inside the matrix, where the basis functions inside the fracture domain are imposed as Dirichlet boundary conditions. To confine the basis functions to their intended local domains, leaked local solutions are redistributed proportionally. Finally, a multiscale finite-volume (MSFV) coarse system is constructed using the two sets of basis functions and then solved. The coarse-scale solution is then interpolated to the fine scale resolution. This MS solution can be used directly as an accurate approximation. Alternatively, the method is paired with a second-stage smoother in order to allow for a convergent and conservative error reduction strategy.

The MS-DFM method proves accurate for a wide range of test cases. Furthermore the additional cost of flexible unstructured grids is expected to be compensated by the efficiency gains from the multiscale approach. Therefore, MS-DFM provides a promising framework for field-scale simulation of flow in fractured media using a DFM approach.

The thesis is structured as follows. First, the governing equations and fine-scale DFM system on unstructured grids are presented. The multiscale finite volume method for unstructured grids is then introduced in Section 3, along with the iterative multiscale solution strategy. The MS-DFM framework is presented in Section 4. Systematic numerical tests are presented to first validate the fine-scale model, and then assess the performance of the MS-DFM. Finally, the paper is concluded in Section 6.

2

Fine-Scale Discretized System

Single-phase incompressible flow in porous media with discrete fractures can be described using Darcy's law as

$$-\nabla \cdot (\lambda \cdot \nabla p) = q, \tag{2.1}$$

where λ, p, and q are the mobility, pressure, and source terms, respectively. Equation (2.1) is solved on an unstructured mesh following the discrete fracture modeling approach, where fractures are represented as lower dimensional features in R^{n-1} space, connected to matrix cells in the R^n space [9].

The matrix grid is generated such that fracture elements are confined to the matrix cell interfaces. Figure 2.1 illustrates an example of such a 3D domain. In this work, the mesh generators TetGen [24] and Triangle [23] are used for 3-dimensional (3D) and 2-dimensional (2D) domains, respectively.

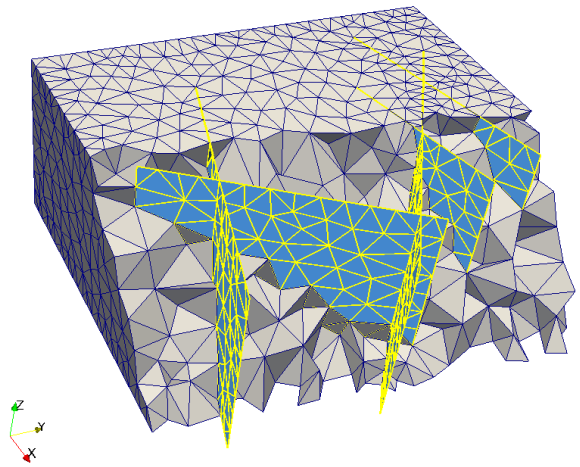


Figure 2.1: View of conforming 3D grid cells.

Virtual fracture control volumes, in virtual R^n space, are constructed by applying the aperture size as the missing dimension of the fracture elements. This allows for a convenient discretization scheme where the flux interactions between the neighboring cells can be expressed in a generic form, irrespective to which domain (fracture or matrix) they belong. Applying a finite-volume method,

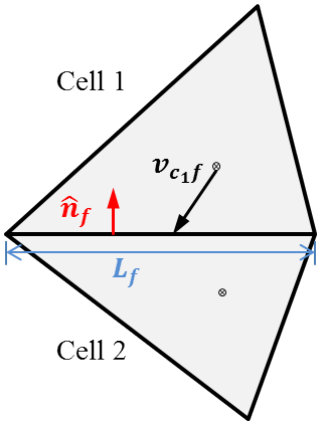


Figure 2.2: Half Transmissibility Variables in a 2D cell

Eq. (2.1) is discretized for every control volume, i.e., matrix and fracture. In the classical two-point flux approximation scheme [28], implemented in this work, the discrete system reads

$$\text{for every cell i:} \quad \sum_{j=1}^{n_{nb}} T_{ij}(p_j - p_i) = q_i, \tag{2.2}$$

where n_{nb} is the number of neighboring cells. In addition, T_{ij} is the transmissibility between cell i and its neighboring cell j, computed as

$$T_{ij} = \frac{T_{if} T_{jf}}{T_{if} + T_{jf}}, \tag{2.3}$$

with the half transmissibility values T_{if} and T_{jf} defined as

$$T_{if} = \frac{A_f * (\vec{n}_f \cdot (\mathbf{k_i} \cdot \vec{n}_f))}{\vec{n}_f \cdot \vec{v}_{cf}}. \tag{2.4}$$

Here, "f" and "c" subindices indicate face and cell quantities, respectively. Accordingly, A_f represents the face area, $\vec{n}_f$ the unit normal vector to the face, $\vec{v}_{cf}$ the vector from the cell center to any point on the face and $\mathbf{k_i}$ the permeability tensor of cell i. Figure 2.2 illustrates these variables for a 2D cell (hence, the area is replaced by the interface length L_f). This TPFA expression was opted for after a comparison between existing formulations. The results of this evaluation are presented in Appendix A.

To avoid small control volumes at fracture intersections, the star-delta procedure is applied [9]. In this case, the sum of all half-transmissibilities acting on the intersection is the denominator of Eq. (2.3). For the transmissibility between a cell i and cell j, this results in

$$T_{ij} = \frac{T_{if} T_{jf}}{\sum_{k \in Fr} T_{kf}}, \tag{2.5}$$

where Fr is the set of fractures connected to the intersection. Note that, here, the face, f, is now the virtual interface at the intersection.

Following the described procedure, the linear system corresponding to Eq. (2.2) finally reads

$$A\ p \equiv \begin{bmatrix} A_{mm} & A_{mf} \\ A_{fm} & A_{ff} \end{bmatrix} \begin{bmatrix} p_m \\ p_f \end{bmatrix} = \begin{bmatrix} q_m \\ q_f \end{bmatrix} \equiv q, \tag{2.6}$$

where subscripts "m" and "f" indicate matrix and fracture quantities, respectively. For example, A_{mf} contains the influence of fractures on the matrix control volumes.

The linear system (2.6) provides an accurate (fine-scale) solution to Eq. (2.1), the accuracy of which will also be illustrated via a test case in Section 5.2. However, real-field simulations require large number of matrix and fracture cells with highly contrasting properties, therefore imposing a significant challenge to computational methods. To resolve this challenge, in this work, a multiscale method is developed to provide an accurate and efficient approximation of the discrete fracture model as presented in Eq. (2.6).

3

MSFV on Unstructured Grids

In this section, the multiscale finite volume method (MSFV) [8] is first extended for the flow (pressure) solution on fully unstructured grids. Such a development has not been addressed in literature. Then, in the next section, it is extended to include discrete fractures within a DFM framework.

MSFV efficiently approximates the fine scale solution by solving a coarse system constructed using local basis functions. While these basis functions are solved on a dual-coarse grid, the method also relies on a primal coarse grid to construct a conservative coarse-scale system. Therefore, an important step in extending the MSFV to unstructured grids is the appropriate construction of primal and dual coarse grids. Previous efforts in the unstructured domain have used either a restricting logic in constructing these grids, or relied on a global iterative procedure for local basis functions [15, 22]. The unstructured MSFV method, proposed in this paper, allows for flexible coarse grids while preserving the simplicity of the original structured method and its algebraic description [30]. In the following subsections, the MSFV method and the iterative solving procedure are described. Then, the primal and dual grid construction are discussed. Finally, the procedure to obtain basis functions is presented.

3.1. MSFV Formulation

The unstructured multiscale method provides an approximate solution to Eq. (2.6), based on a superposition expression which can be algebraically stated as

$$p \approx p' = \mathbf{P}\check{p}. \tag{3.1}$$

Here, the fine-scale solution p is approximated by the multiscale solution p' which is obtained by interpolation (prolongation) of the coarse-scale solutions $\check{p}$ and the matrix of basis functions $\mathbf{P}$ (prolongation operator). The coarse-scale pressure $\check{p}$ is found by solving the coarse-scale system $\check{A}\check{p} = \check{q}$, which is constructed algebraically as

$$\underbrace{(\mathbf{R}\,A\,\mathbf{P})}_{\check{A}}\,\check{p} = \underbrace{\mathbf{R}q}_{\check{q}}. \tag{3.2}$$

Here, $\mathbf{R}$ and $\mathbf{P}$ are the restriction and prolongation operators, respectively. The restriction operator $\mathbf{R}$ takes the fine-scale solution to the coarse scale, resulting in a reduction of unknowns. For a finite-volume-based multiscale formulation (MSFV), the restriction operator represents the integration operator over the coarse-grid cells, i.e., for its entry $R(i,j)$ one can state

$$R(i,j) = \begin{cases} dv_j & \text{if fine-cell } j \text{ belongs to coarse cell } i \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

where dv_j is the volume of the fine cell j. Therefore, $\mathbf{R}$ is a $N_c \times N_f$ matrix where N_c and N_f are the number of coarse and fine cells, respectively [25, 30].

The prolongation operator $\mathbf{P}$ consists of the basis functions, Φ, which are placed in each of its columns, i.e.,

$$\mathbf{P} = \begin{bmatrix} \vdots & & \vdots & & \vdots \\ \Phi_1^m & \cdots & \Phi_i^m & \cdots & \Phi_{N_{mc}}^m \\ \vdots & & \vdots & & \vdots \end{bmatrix}_{N_f \times N_c} . \tag{3.4}$$

In brief, basis functions are local interpolators which capture the complex fine-scale solution in the vicinity of a coarse node. There exists one basis function for each coarse node, and their local support region is defined based on the dual-coarse grid. These basis functions are explained in more detail later, once the construction of the primal and dual coarse grids have been described. Before doing so, next, the iterative MSFV procedure is presented.

3.2. Iterative MSFV procedure

The multiscale solution p' can be used directly as an efficient approximation of the reference solution. However, to obtain a more precise solution, especially for highly heterogeneous problems, the iterative MSFV method can be applied to converge the multiscale approximation to the fine-scale reference solution [5, 16]. To this end, the unstructured multiscale method is combined in a multi-stage iterative procedure, in which a number of fine scale smoother iterations, n_{smooth}, is applied to ensure convergence.

At each smoother iteration a smoother operator, $\mathbf{M}^{-1}$, is applied. In this work, this operator is constructed via ILU(0) decomposition, approximating the inverse of the fine scale linear operator A. The procedure is iteratively repeated until the specified residual norm is reached. An overview is provided in Alg. 1.

Algorithm 1 Iterative multiscale strategy. Here, ϵ is the desired threshold.

$\nu = 1$
$p^\nu = 0$ Initiate
$r^\nu = q - Ap^\nu$ Compute residual

while $(||r||_2^\nu > \epsilon)$ **do**
  Multiscale stage:
  $\check{r}^{\nu+1/2} = \mathbf{R}\, r^\nu$ *Compute coarse residual*
  $\delta\check{p}^{\nu+1/2} = \check{A}^{-1}\, \check{r}^{\nu+1/2}$ *Compute coarse pressure correctiom*
  $\delta p^{\nu+1/2} = \mathbf{P}\delta\check{p}^{\nu+1/2}$ *Prolong to find fine pressure correction*
  $p^{\nu+1/2} = p^\nu + \delta p^{\nu+1/2}$ *Update pressure*
  $r^{\nu+1/2} = q - Ap^{\nu+1/2}$ *Update residual*

  Smoother stage:
  $p^{\nu+1} = p^{\nu+1/2} + \mathbf{M}_{ILU(0)}^{-1} r^{\nu+1/2}$ *Apply Smoother*
  $r^{\nu+1} = q - Ap^{\nu+1}$ *Update residual*

  $\nu \leftarrow \nu + 1$
end while

3.3. Primal and Dual Grid

To construct the restriction and prolongation operators, a primal and dual coarse grid need to be constructed. The primal grid represents the coarse grid, or partitioning, and is constructed using METIS [10] in this work. To allow for full flexibility, the only input is the fine scale connectivity. Future research will address coarse grids which can improve the performance of the MS-DFM method by using specified partitioning criteria or weights, e.g., permeability.

Based on the primal partitioning, the dual grid construction follows the hierarchy of its components: dual-nodes, dual-edges, dual-faces and internal cells. Figures 3.1 and 3.2 depict a primal- and its corresponding dual-coarse cell for a 2D test case.

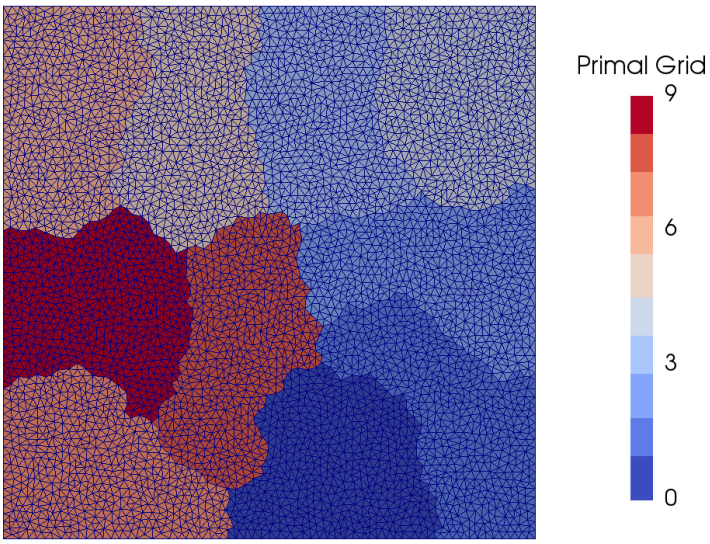


Figure 3.1: Coarse grid imposed on a 2D unstructured fine grid obtained by METIS [10].

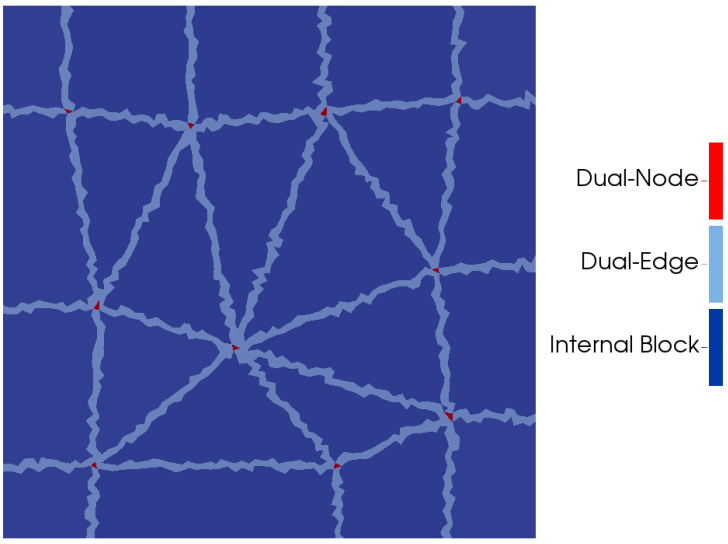


Figure 3.2: Dual-Grid corresponding to the partitioning of Fig. 3.1.

To construct the dual-grid, first, a fine cell is selected as the coarse node (dual-node) inside each coarse cell. In this work, the fine cell closest to the volume weighted center is chosen as the coarse node.

Then, the dual-edge cells are selected such that they connect coarse nodes (dual-nodes) to their neighboring coarse nodes. Neighbors are defined based on the primal coarse grid connectivity. Pointing from the initial dual-node to the target dual-node, the target vector facilitates the construction of the dual-edges. Guided by this target-vector, fine cells are added to the dual-edge until the dual-nodes are connected. By flagging the face of the dual-edge cell which is intersected by the target vector, the neighboring cell is identified as the next dual-edge cell. This process is summarized in Alg. 2.

Algorithm 2 Construction of Dual-Edges

```
for i = 0 to n_dualnodes do
    for all j = connected dualnodes do
        CurCell = IniCell                          ▷ IniCell is the fine cell of dual-node i
        while CurCell ≠ TargetCell do
                                                   ▷ TargetCell is the fine cell of dual-node j
            Assign CurCell to the dual-edge
            Find intersected interface of CurCell
            CurCell ← neighbor of intersected interface
        end while
    end for
end for
```

Next, dual-faces are naturally constructed between dual-edges. To identify dual-face cells, all grid nodes are first projected onto the face-plane. This plane is constructed by the (typically 3) connected dual-nodes which construct the dual-face. Grid nodes projected inside of the virtual face (i.e. the area between the connected dual-nodes) are then assigned a code of 1 if they are above the plane or a code of 2 if they are below the plane. Above and below are defined with respect to the direction of the computed normal vector of the face-plane, i.e.,

$$Code = \begin{cases} 1 & \text{if } (\vec{n}_{fp} \cdot \vec{v}) > 0 \\ 2 & \text{otherwise,} \end{cases} \tag{3.5}$$

where $\vec{n}_{fp}$ is the normal vector to the face-plane and $\vec{v}$ is the vector from a grid node's projection to the grid node itself. The remaining nodes, with projections outside the virtual face, are assigned the code of 0. Once all grid nodes are coded, dual-face cells can be identified. Fine cells which are constructed by at least one grid node with code 1 and one grid node with code 2 are assigned to the dual-face. This implies that a dual-face cell crosses the face-plane. If a cell is already part of a dual-grid component of higher hierarchy, i.e., a node or edge, it remains unchanged. This process is recapitulated in Alg. 3, where $CellNodeCodes$ is the set of codes present in the grid nodes constructing the fine cell.

Finally, the internal blocks are defined. All cells not yet assigned to a dual-grid structure are naturally internal cells. However, to solve strictly local problems when obtaining basis functions (see 3.4), cells are assigned a unique code per block. To do so, a flood-fill algorithm is implemented (Alg. 4). First, it finds an unassigned cell and initiates the construction of a new block. The block is then expanded with all unassigned neighbors of the initial cell. This is repeated for all newly added cells until all neighbors are already assigned to higher ranked dual-grid components. This process is then reiterated until all internal blocks are numbered. Small clusters of internal cells, which are separated from the main block by meandering dual-edges and dual-faces, are added to the neighboring structure with the lowest level of hierarchy, i.e., dual-face preferred over dual-edge.

Algorithm 3 Construction of a Dual Face

```
for i = 0 to n_gridnodes do
    Project node on Face Plane
    if Projection inside virtual face (check with barycentric coordinates) then
        if Node is above face plane (see Eq. (3.5)) then
            Node assigned code 1
        else
            Node assigned code 2
        end if
    else if Projection outside virtual face then
        Node assigned code 0
    end if
end for

for all i = UnassignedCells do
    if {1,2} ∈ CellNodeCodes then
        Cell i assigned to the dual-face
    end if
end for
```

Algorithm 4 Construction of Internal Blocks

```
for i = UnassignedCells do
    Initiate new block
    Assign cell i to block
    Initiate list CurrentCells with i
    while n_CurrentCells > 0 do

        for all j = CurrentCells do
            for all k = Unassigned neighbors of j do
                Assign cell k to block          ▷ cell k removed from UnassignedCells
                Add cell k to NewCells
            end for
        end for

        CurrentCells ← NewCells

    end while
end for
```

Note that dual-edges and dual-faces may *overlap* with other dual-edges and dual-faces, respectively. These overlapping features create edge-networks and face-networks which are critical when solving for basis functions (see 3.4). The edge networks of the previously displayed 2D test case are shown in Fig. 3.3.

Dual-edges may also *cross* with other dual-edges or dual-faces. A crossing dual-edge doesn't share any common connected dual-nodes with the crossed structure. This differs to when it *overlaps* with a dual-edge or neighbors a dual-face. When a crossing dual-edge is identified, the dual-edge is deleted. As a consequence, dual-faces constructed based on the deleted dual-edge are removed as well. This happens during the construction process and therefore does not influence the internal block algorithm which takes place afterwards. Note that due to these deletions, edge- and face-networks may be segmented and therefore will be treated as multiple separate networks when computing basis functions.

Also note that the described dual grid construction routine does not take into account any non-geometric information. This allows for broad application. Nevertheless, the addition of rules based on physical properties may allow for more accurate and robust multiscale solutions as well as improved convergence in iterative solving strategies. This is the subject of ongoing research. Furthermore, the described procedures were found to be efficient and consistent in constructing dual grids that satisfy the overall MSFV requirements. However efficiency improvements will be investigated in future work.

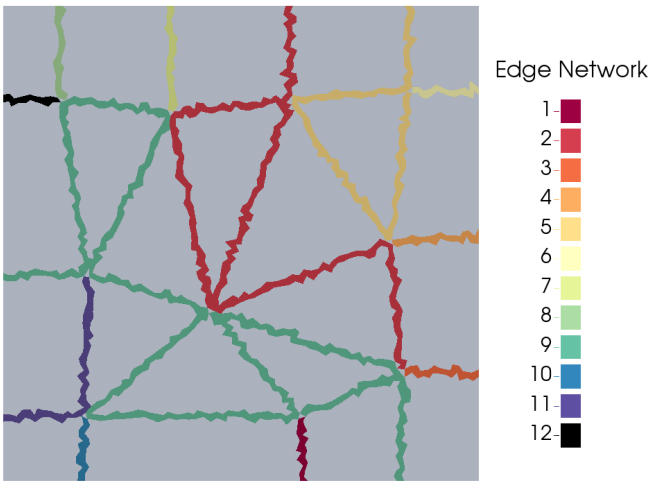


Figure 3.3: Edge networks in 2D example of Fig. 3.2.

3.4. Basis Functions

Physics-based basis functions are obtained by solving the governing equation, Eq. (2.1), on local dual-coarse blocks without right hand side terms. To do so, reduced dimensional boundary conditions are imposed in the way that first the dual-edge domains are solved using Dirichlet values of 0 and 1 at the coarse nodes. The obtained solutions serve as Dirichlet condition for the dual-face domain. Finally, the internal cells are solved subject to a Dirichlet condition from all higher-ranked cells. The procedure is summarized in Alg. 5.

Due to overlapping structures, solving basis functions per component (i.e., per dual-edge) as done in MS on structured grids is not possible. Instead, the algorithm solves per edge network, i.e., cluster of relevant connected edges. Likewise, basis functions are obtained by solving over face networks rather than individual dual-faces.

Algorithm 5 Basis function solving procedure

for $i = 0$ to $n_{edgenetworks}$ **do**
  for all j = influencing dual-node **do**
    Set value Cell j = 1
    Set value other influencing dual-nodes = 0
    Solve basis function for edge network
  end for
  Rescale solution for edge cells in edge-network i (Eq. (3.6))
end for

for $i = 0$ to $n_{facenetworks}$ **do**
  for all j = influencing dual-node **do**
    Set neighboring dual-edge and dual-node cells to ϕ_j value
    Solve basis function for face network
  end for
  Rescale solution for face cells in face-network i (Eq. (3.6))
end for

for $i = 0$ to $n_{internalblocks}$ **do**
  for all j = influencing dual-node **do**
    Set neighboring dual-face, dual-edge and dual-node cells to ϕ_j value
    Solve basis function for internal block
  end for
  Rescale solution for internal cells
end for

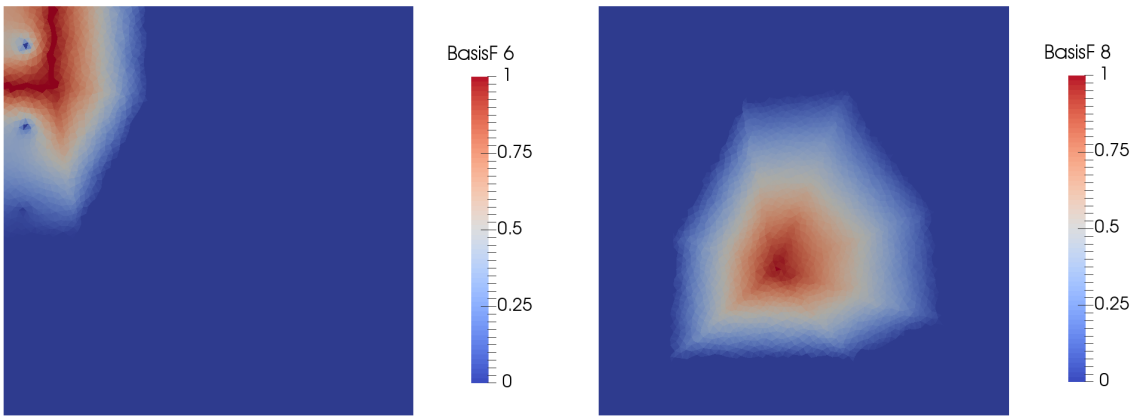
As a side-effect of solving over edge- and face-networks, basis function solutions can leak outside the area of influence of their dual-node. This area of influence represents the direct region around the dual-node, and consists of all cells which are part of a child structure of that dual-node. Here a child structure is a dual-grid component created due to connections of the dual-node. To confine the basis functions to the area of influence of their dual-node, the solutions are rescaled. Rescaling entails setting all basis function values that should not contribute to the evaluated cell to zero, and redistributing the subtracted values proportionally among the remaining basis function contributions. In this way the partition of unity is preserved. This can be written as

$$\text{for cell i: } \mathbf{P}_{ij} = \begin{cases} \frac{\phi_{ij}}{\Sigma\phi_i} & \text{if basis function j} \in \mathbf{bf}_i \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\mathbf{bf}_i$ is the family of basis functions contributing to cell i, and $\sum\phi_i = \sum_{j=0}^{\mathbf{bf}_i}\phi_{ij}$ is the sum of basis function values in $\mathbf{bf}_i$ for cell i.

Rescaling according to the area of influence in the internal blocks could result in loss of important information in the basis functions. For example, in the case when elongated primal blocks result in largely overlapping edges and faces. However, to reduce the amount of non-zeros in $\mathbf{P}$ and more importantly in $\check{A}$, the local solutions in the internal blocks are rescaled according to thresholds, e.g., a minimum contribution to a block. It is important to note that although rescaling for confinement leads to a more robust simulator, it can also lead to slightly increased error if an iterative solving strategy is not employed. However, with iterations, confinement also leads to increased efficiency.

The presented fully unstructured procedure increases the applicability of MSFV methods significantly, and allows them to be employed for the simulation of more complex geological features compared to their existing scope. Although one can apply the method to DFM directly, i.e., by taking into account the virtual fracture control volumes, the efficient and accurate simulation of these highly contrasting features demands a more specialized approach.



(a) (b)

Figure 3.4: Basis functions of dual-node 6 (left) and 8 (right). Basis function 6 is also influenced by 2 well basis functions [31].

4

MSFV for Discrete Fracture Modeling on Unstructured Grids

Fractures demand appropriate treatment within the multiscale framework [16]. To this end, this section presents the extension of the unstructured MSFV method to incorporate the discrete fracture modeling approach (DFM). The proposed MS-DFM method constructs independent primal grids for the matrix and fractures. As such, it generates independent dual-grids and therefore does not require additional coarse grid-related computations in the matrix domain. In this section, first the dual coarse grid construction for fractured media is described. Then, the basis function formulation and appropriate fracture-matrix coupling is presented.

4.1. Dual Grid in Fracture Domain

As mentioned, the fracture and matrix coarse grids are totally independent. Therefore the same grid generation strategy as previously described for non-fractured media is employed for the matrix. For the lower dimensional fractures, the dual grid construction follows a similar procedure. More precisely, only the fracture dual-edge routine differs from that of the matrix. The fracture dual-edges are again built cell-by-cell; however, a direction-based criterion is used to determine the next dual-edge fracture cell. This alters Alg. 2, line 6. The algorithm evaluates the criterion over each of the fracture cell's edges (i.e. interfaces, not to be confused with dual-edge) and can be described as follows. Starting from a coarse node, the projection of the vector extending from the current fine cell to the target coarse node is named vector B. The vectors from the current cell center to the nodes of the assessed cell-edge are named A and C. Figure 4.1 illustrates the vectors described. The code then evaluates the following set of conditions:

$$\text{if } ((A \cdot B < 0) \;\;\&\;\; (C \cdot B < 0)) \text{ then the edge is in the wrong direction} \tag{4.1}$$

and

$$\text{if } (((A \times B) \cdot (A \times C) < 0) \;\;\&\;\; ((C \times B) \cdot (C \times A) < 0)) \text{ then B intersects the edge.} \tag{4.2}$$

If an edge (i.e., interface in 2D) is in the right direction and is intersected by B, then the new dual-edge cell is the neighbor of this interface.

Fracture intersections, where two or more fracture plates intersect, present an additional challenge for the dual-grid construction in the fracture domain. When a dual-edge connects two dual-nodes on different plates, the dual-edge construction algorithm first targets the to-be-taken fracture intersections. Once at the intersection, the dual-edge switches plates and the algorithm aims at for

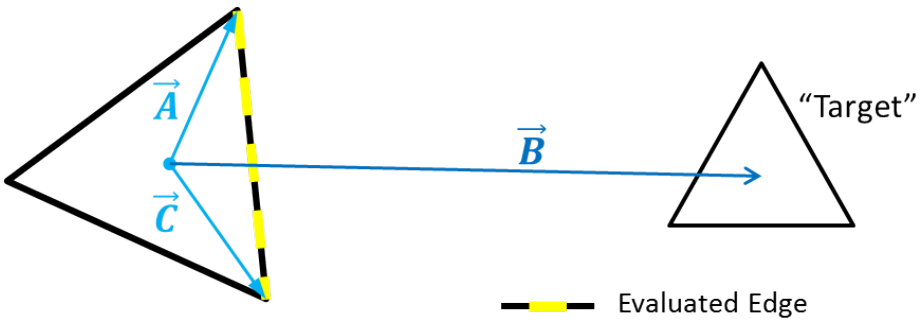


Figure 4.1: Illustration of the vectors used when evaluating an edge.

following target. Moreover, a common intermediate target at a fracture intersection for all connected dual-nodes –crossing the same fracture intersection– can be advantageous. In combination with enlarged areas of influence(see 4.2), this common target connects all surrounding dual-nodes to each other, which provides a more accurate and realistic solution when fracture plate properties are heterogeneous.

4.2. Basis Functions & Matrix-Fracture Coupling

As both fracture and matrix dual-nodes are present, the multiscale superposition expression for non-fractured media, presented in Eq. (3.1), is extended to include fractures. Here, the basis functions in the fracture domain Φ^f are solved independently, i.e., with no consideration of matrix connectivity. Therefore, their formulation is a straightforward localized version of the governing equation on lower-dimensional manifolds. No-flow conditions are imposed at their connections with the matrix domain. Consequently, the multiscale superposition expression in the fracture domain reads

$$p'^f = \sum_{i=1}^{N_{cf}} \Phi_i^f \check{p}_i^f, \tag{4.3}$$

where $\check{p}^f$ stands for the fracture coarse-scale pressures and N_{cf} for the coarse-scale fracture unknowns (fracture dual-nodes). Additionally, around fracture intersections the area of influence is enlarged such that Φ^f can be influenced by all neighboring fracture dual-nodes, which are connected to the intersection.

While the basis functions inside the fracture domain are solved independently (with respect to the matrix), the matrix basis functions are influenced by the two media. Therefore, the set of matrix basis functions includes those basis functions which capture the effect of matrix itself, i.e., $\Phi^{m,m}$, and those which capture the effects of fractures, i.e., $\Phi^{m,f}$. This means that Fracture-Matrix coupling only occurs from fracture to matrix and not vice-versa (see [16] for more information), which leads to the

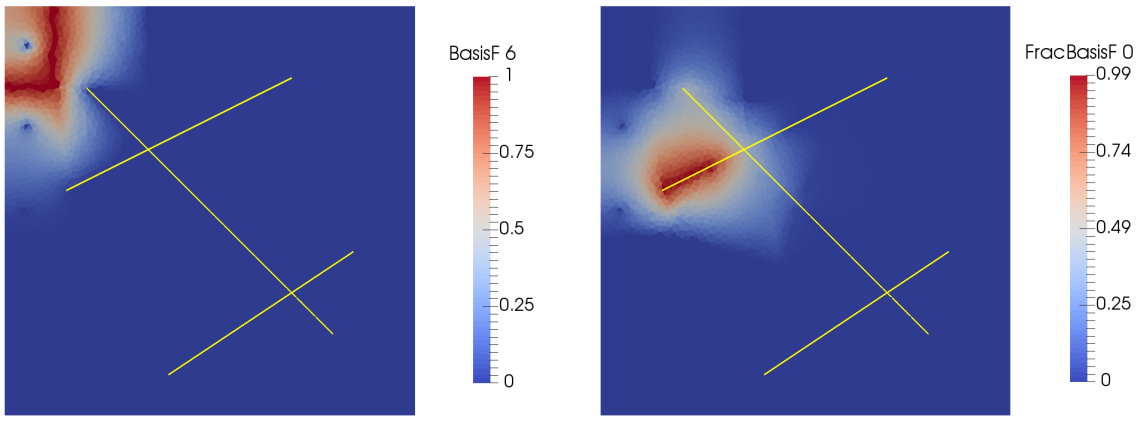
$$p'^m = \sum_{i=1}^{N_{cm}} \Phi_i^{mm} \check{p}_i^m + \sum_{i=1}^{N_{cf}} \Phi_i^{mf} \check{p}_i^f \tag{4.4}$$

expression. Here, $\check{p}^m$ indicates the coarse-scale matrix pressure, and N_{cm} is the total count of matrix coarse-scale pressure unknowns. Note that coupling in both directions is possible, i.e., fractures could also account for the influence of the matrix cells, however, it would lead to a denser prolongation operator and more constrained dual-grid generation.

To account for the fracture basis functions in the matrix domain, fracture basis functions Φ^f act as Dirichlet boundary conditions for the Φ^{mf} basis functions. In these local problems all matrix

dual-nodes are set to zero. Vice-versa, matrix-matrix basis functions Φ^{mm} now have additional boundary conditions. Namely, Dirichlet values of zero are imposed in connected fracture cells.

Rescaling for confinement (see Eq. (3.6)) still follows the same principles. The area of influence of the fracture basis functions in the matrix domain is defined by all the matrix structures which are directly connected to a fracture cell influenced by that basis function. Figure 4.2b shows such a fracture basis function in the matrix domain. Note that this is the same test case as in Figs. 3.1-3.4 where now the cell faces indicated by the yellow line are activated as highly conductive fractures. Figure 4.2a shows the effect on the previously presented matrix basis function and Fig. 4.3 illustrates the total effect of all fracture basis functions in the matrix domain, i.e., $\sum \Phi^{mf}$.



(a) (b)

Figure 4.2: (a) Basis function of a dual coarse node 6, i.e. Φ_6^{mm}, with fractures being set to a Dirichlet value of 0. (b) The basis function of fracture dual-node 0 in the matrix domain, i.e., Φ_0^{mf}.

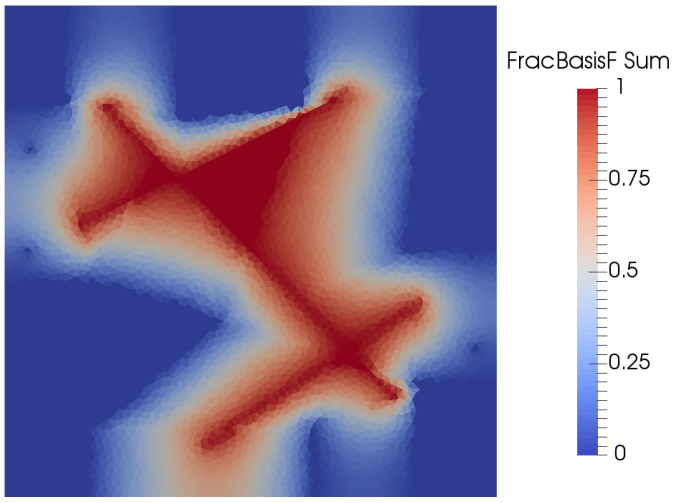


Figure 4.3: Illustration of the effect of $\sum \Phi^{mf}$, i.e., the sum of all fracture basis functions in the matrix domain.

4.3. Multiscale Operators

The restriction operator **R** is naturally enlarged with rows containing the fracture primal grid partitioning volumes. The coupling of the fracture and matrix domain leads to an extended prolongation operator,

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^m \\ \mathbf{P}^f \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{mm} & \mathbf{P}^{mf} \\ 0 & \mathbf{P}^{ff} \end{bmatrix}, \tag{4.5}$$

where, more precisely,

$$\mathbf{P}^m = \begin{bmatrix} \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{mm} & \cdots & \Phi_{N_{cm}}^{mm} & \Phi_1^{mf} & \cdots & \Phi_{N_{cf}}^{mf} \\ \vdots & & \vdots & \vdots & & \vdots \end{bmatrix}. \tag{4.6}$$

As presented, the coarse system for the MS-DFM model is a straightforward extension of the non-fractured model where the prolongation and restriction operator are extended naturally. Though only coupled in one direction in the local problems, this method leads to a fully coupled coarse scale system.

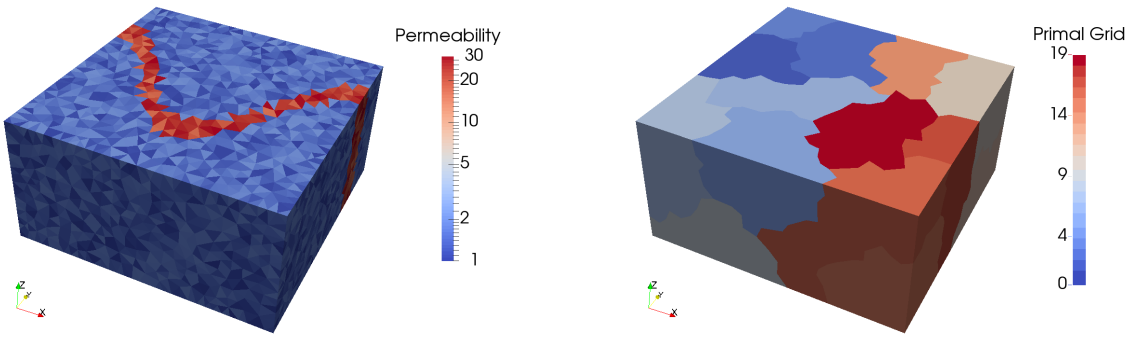
5

Numerical Results

To verify, validate and assess the performance of the MS-DFM model, this chapter presents numerical results of a series of test cases: (1) an unstructured multiscale verification test case, (2) a 2D DFM validation test case, and (3) 2D and (4) 3D MS-DFM test cases. Furthermore, appendix B presents an additional fine scale test case which was designed to verify the recently developed pEDFM [27].

5.1. MS on unstructured grids: verification

First of all, the multiscale finite volume method for non-fractured porous media on unstructured grids is validated. A 3-dimensional test case is devised on a $1[m] \times 1[m] \times 0.5[m]$ matrix domain which is perforated by 12 wells [18]. The wells are in a line drive pattern with 6 injecting perforations at $x = 0.05$ positioned in two rows of 3 at $z = 0.05$ and $z = 0.45$. The producing perforations are in an identical pattern at $x = 0.95$. The field is assigned a random permeability field with a high perm channel meandering through the domain, as shown in Fig. 5.1a. The fine scale grid consists of 19882 cells, which is partitioned into 20 coarse cells for the MS coarse grid. The primal and dual grid are illustrated in Figs. 5.1b and 5.2a, respectively, where the dual grid is illustrated by means of the internal blocks which are separated by the higher-ranked structures (dual-edges, dual-faces and dual-nodes).



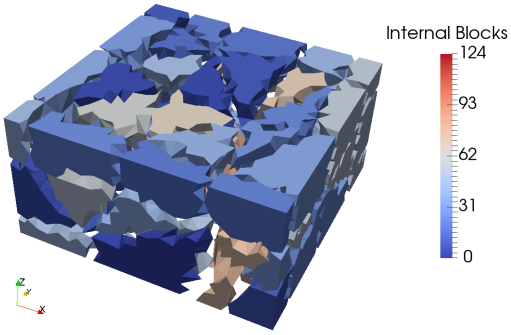
(a) Permeability map

(b) Coarse Grid

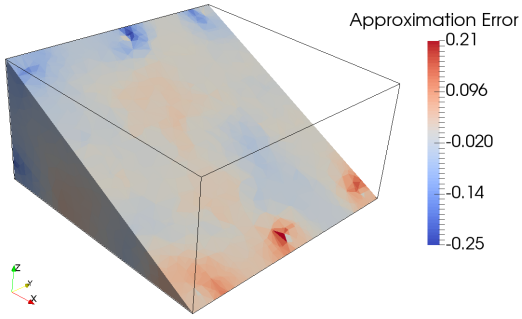
Figure 5.1: Illustrations of the problem set-up and primal coarse grid.

The fine scale and multiscale solutions are depicted on a diagonal cross section of the matrix domain in Figs 5.2c and 5.2d, respectively. Note that the color scale has been adapted to emphasize

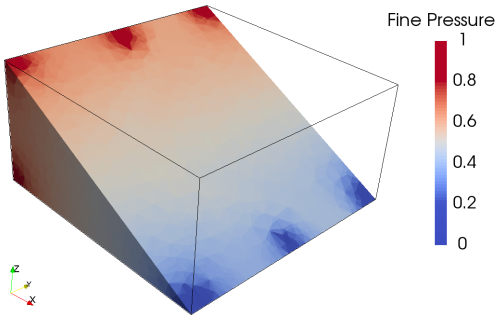
the non-dimensional pressure values between $p = 0.2$ and $p = 0.8$. The error between these solutions is indicated in Fig. 5.2b. From the error plot it is clear that the peaks of the discrepancies lie close to the wells. This is especially the case when dual-edges or dual-faces pass nearby the well but do not incorporate the perforated cell. These spikes occur because of the well-known localization error introduced at the boundary of the basis functions. Despite the spikes, the multiscale solution approximates the fine solution accurately with a maximum discrepancy of 0.07 in the rest of the domain. Note that by applying iterations of i-MSFV procedure, one can systematically reduce the error to any desired level.



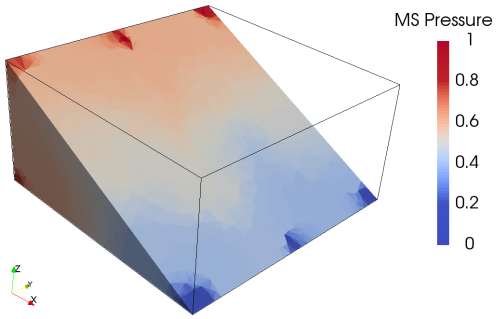
(a) Internal Blocks of Dual-Grid



(b) MS-DFM error



(c) Fine-scale



(d) MS-DFM (without iterations)

Figure 5.2: Dual-grid visualization and comparison of the fine-scale reference and MS-DFM solution (without iterations).

5.2. Fine-scale DFM model: validation

Before testing the MS-DFM method for fractured media, the underlying fine-scale DFM model is validated with respect to a fully-resolved reference model. The test case, presented in Fig. 5.3, consists of a $1[m] \times 1[m]$ homogeneous matrix with a set of 3 conductive fractures. The reference model consists of 810×810 cells such that the grid resolution allows for fully resolving fractures with the aperture of $h \approx 0.001235$. To verify the convergence of DFM to the fully resolved case, 5 levels of DFM grid refinement have been employed: 2501, 10003, 40006, 160000 and 640000 cells.

The pressure results of the reference solution and DFM model with 10003 cells are presented in Figs. 5.4a and 5.4b. It is clear that the DFM model accurately captures the real solution. In more detail, Fig. 5.5 plots the average $\mathbb{L}^2$ norm of the error vector (averaged over 100 fixed locations in the domain, after linear interpolation). It is clear from the graph that the error converges with refinement. Thus, the implemented fine-scale DFM model is validated, and will hereafter be referred to and applied as the fine-scale reference model for MS-DFM.

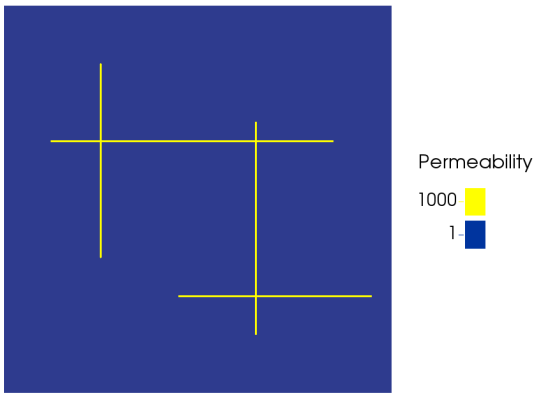
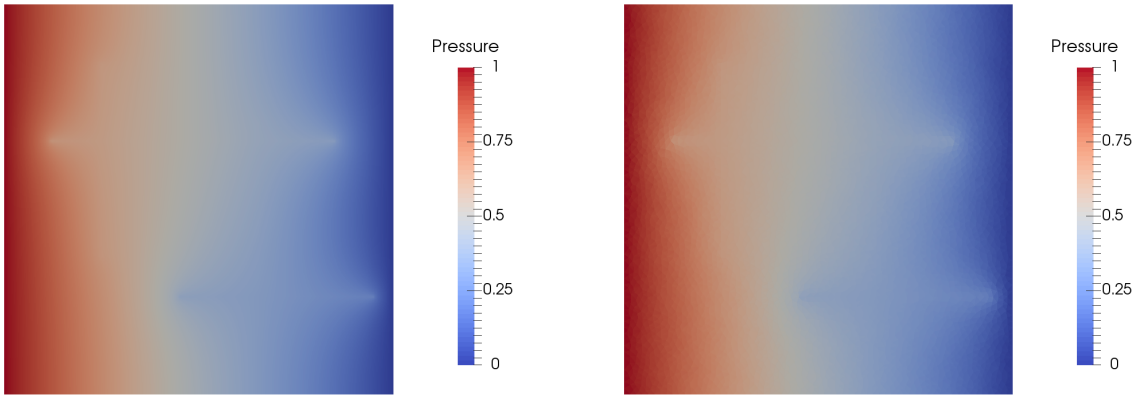


Figure 5.3: The fracture-configuration used for the validation of the DFM model (fractures are emphasized for clarity).



(a) Fully-resolved solution (b) DFM solution

Figure 5.4: Comparison of fully resolved and DFM solutions. Note that the fully resolved case employs 656100 cells, while the DFM employs 10003 cells.

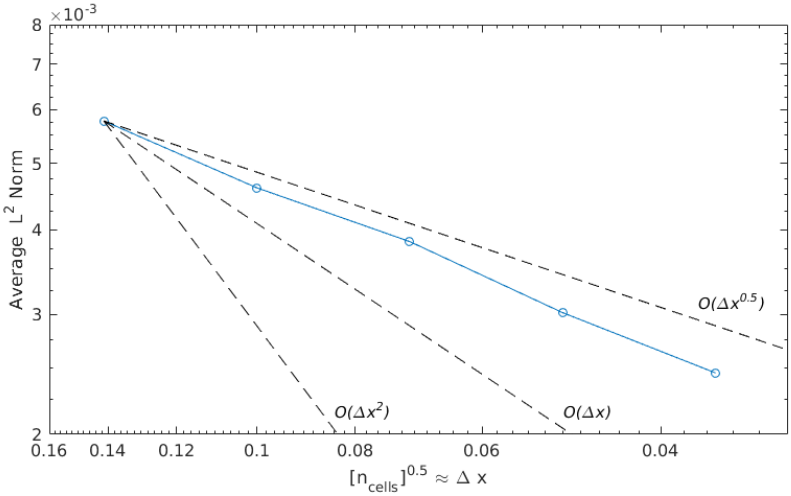
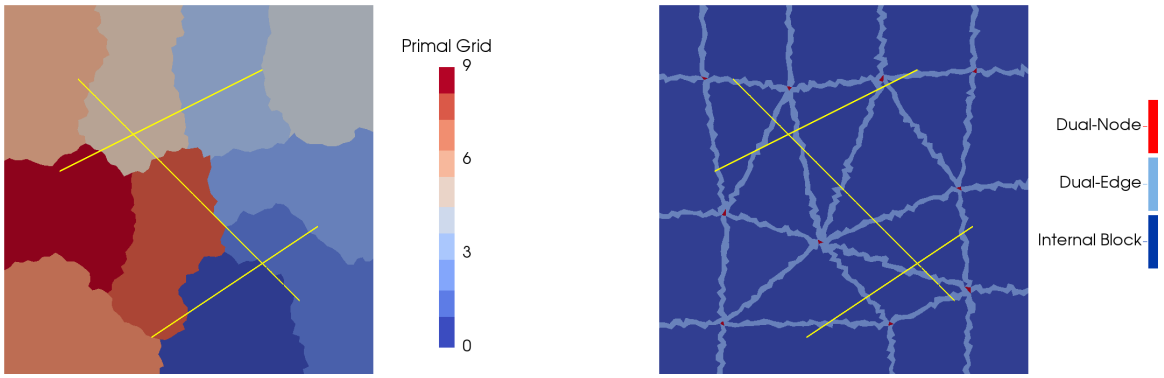


Figure 5.5: Grid convergence of DFM (measured in $\mathbb{L}_2$ norm at 100 fixed locations in the domain) with respect to the fully resolved simulation.

5.3. MS-DFM: 2D test case

A 2D validation test case, in a $1[m] \times 1[m]$ domain, with 3 fractures is considered. The fractures are highly conductive, i.e., $k_f = 1000 * k_m$, and their set-up corresponds with the example shown in Chapter 4. Furthermore, the domain is subject to a line drive of 5 wells on both sides. Injectors, $p_w = 1$, are placed on the left hand side and producers, $p_w = 0$, on the right [18]. The fine scale grid has 10001 matrix and 129 fracture cells where the coarse grid has only 10 matrix and 5 fracture cells respectively. This rough coarsening ratio is a good test for the method's performance. Figures 5.6 and 5.7a present the primal grids of both the fracture and matrix domains as well as the matrix dual-grid. Figures 5.7c and 5.7d present the fine and multiscale (with no iterations) results of the test case.



(a) Primal Coarse Grid (b) Dual Coarse Grid

Figure 5.6: Coarse grids for the 2D Test Case including indicative fractures(yellow)

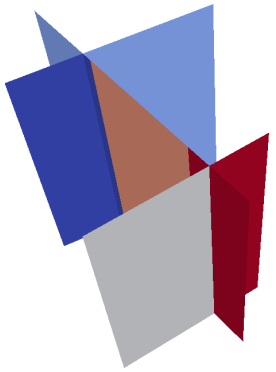
From the multiscale result it is clear that MS-DFM accurately captures the trend of the fine scale solution. Fig. 5.7b more accurately identifies the locations of the mismatch. The pattern of the error closely resembles the dual grid, where the largest discrepancies are naturally detected along edges near boundary conditions and fractures. This is inherent to the multiscale method as the lower dimensional problem along an edge fails to capture nearby boundary conditions. Nevertheless, note that in this case the fracture solution is captured quite accurately as the extremities of fractures intersect dual-edges. As such, the solution with iterations converges to $|r|^2 = 10^{-4}$ after 74 iterations.

5.3.1. MS-DFM: 2D test case, coarse grid refinement

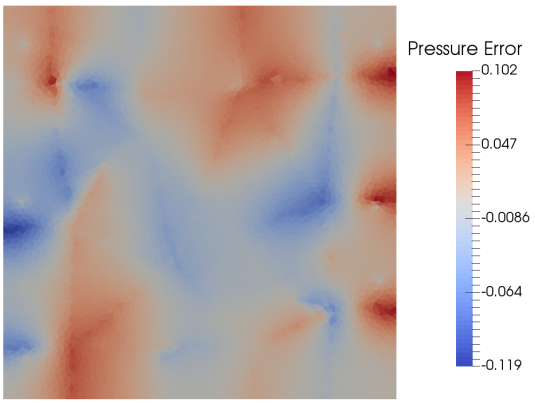
An important aspect of multiscale methods is the coarsening ratio ($n_{finecells}/n_{coarsecells}$). However, as shown in the previous test case, the location of wells and fractures with respect to the dual-grid edges has strong influence on the MS-DFM solution without iterations. Therefore, in contrast to traditional expectations, a lower coarsening ratio is not always a guarantee for a more accurate solution.

To demonstrate this effect, a new 2D test case is devised with 5 fractures on a 1×1 domain. Both media, fractures and matrix, are homogeneous. Furthermore, fractures are assigned highly conductive properties, i.e., $k_f = 1000 * k_m$. The domain is gridded with 10026 fine matrix cells and 170 discrete fracture cells. The domain boundary conditions are 10 wells in a line drive formation with 5 injectors on the left and 5 producers on the right.

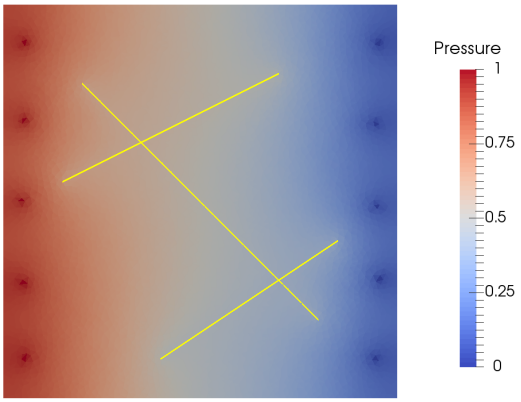
Three levels of coarse grid refinement are tested: (a) 5, (b) 12 and (c) 20 coarse matrix cells with 2, 8 and 18 coarse fracture cells, respectively. Figure 5.8 presents the coarse matrix dual-grid for these cases. Figure 5.9 provides additional insight by presenting the sum of the fracture basis functions in the matrix domain, i.e., $\sum \Phi^{mf}$. Here, in the case with 12 coarse matrix cells, the dual-edge



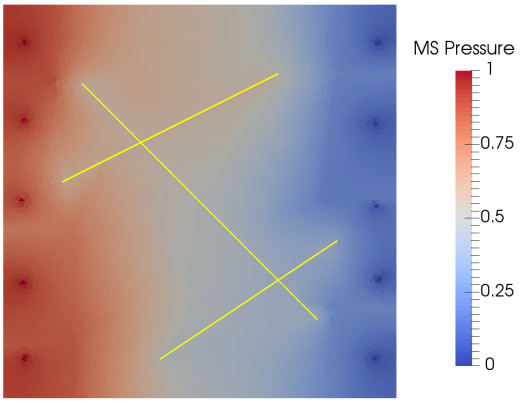
(a) Fracture Primal Coarse Grid



(b) MS-DFM error (no iterations)

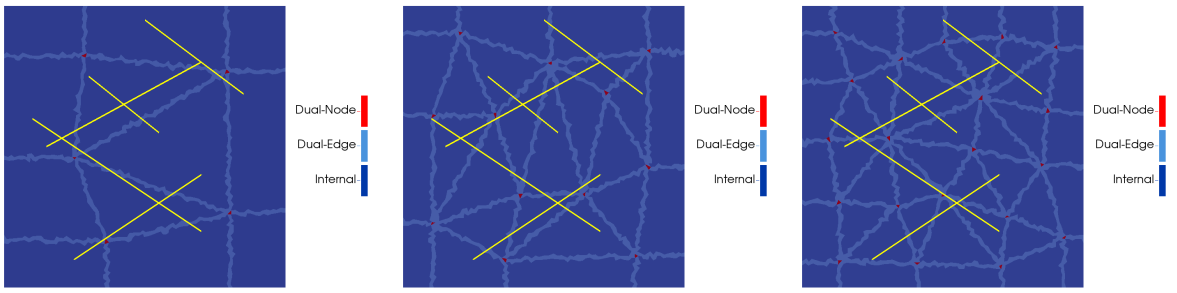


(c) Fine Reference



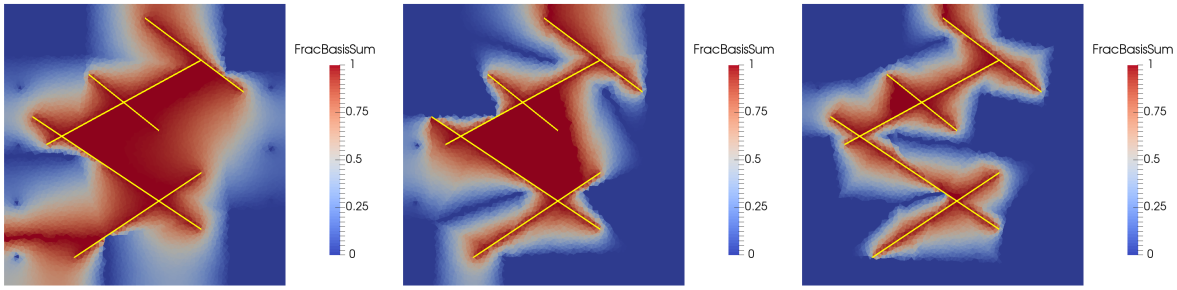
(d) MS-DFM (no iterations)

Figure 5.7: Subfigure (a) depicts the fracture coarse grid on an extruded view of the fractures. Next, (b)-(d) show the comparison of the fine reference and MS-DFM solution for the 2D Test Case.



(a) (b) (c)

Figure 5.8: Matrix coarse dual-grids for 3 levels of coarse grid refinement: (a) 5, (b) 12, and (c) 20 coarse matrix cells.



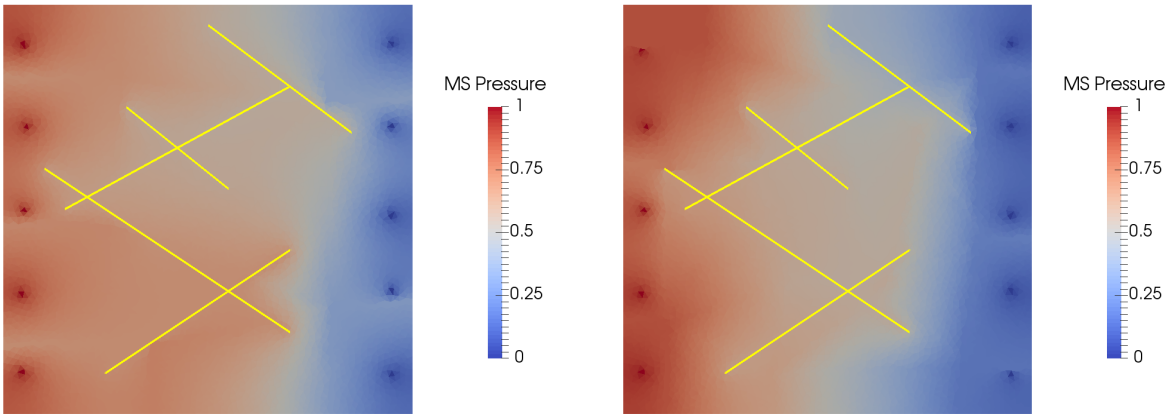
(a) (b) (c)

Figure 5.9: Comparison of the sum of fracture basis functions in matrix domain, i.e., $\sum \Phi^{mf}$, for 3 levels of coarse grid refinement: (a) 5, (b) 12, and (c) 20 coarse matrix cells.

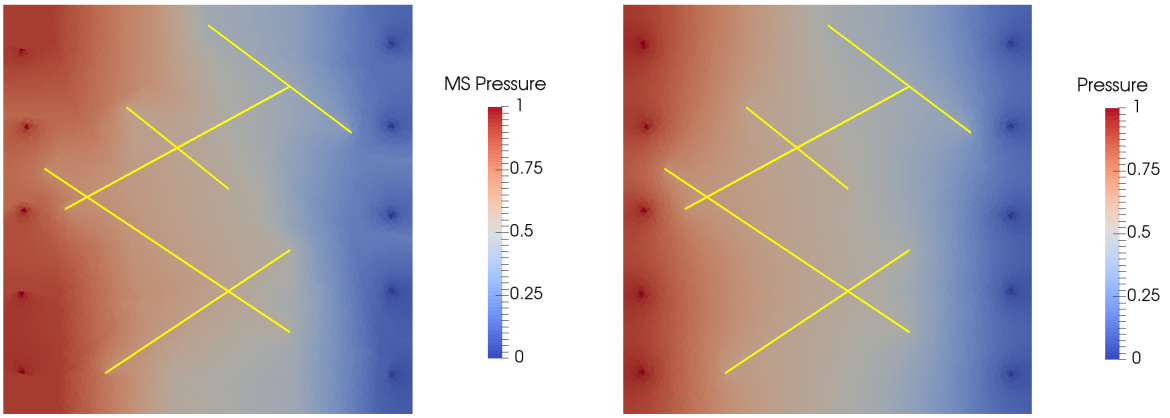
cells between the fractures are not influenced by any matrix dual-nodes (see 5.9b). This naturally results in a basis fracture sum of 1 in this area. Additionally, the area outside the fracture network is relatively poorly influenced compared to the other cases. As a result, the multiscale approximation (without iterations) is notably less accurate. More precisely, the $\mathbb{L}^2$ norm of the residual vector for case (b) is $|r|^2 = 1.645$, while $|r|^2 = 0.870$ and $|r|^2 = 1.231$ for case (a) and (c) respectively.

The pressure results for these cases are illustrated in Figs. 5.10a - 5.10c. The fine scale solution is presented in Fig. 5.10d for comparison. Again, case (b) displays anomalies, showing a blue streak along the bottom of the domain and an abrupt pressure jump at the right hand side of the fracture network. This jump naturally follows from the absence of matrix influence between the fractures.

Note that in this test, we only evaluate the multiscale approximation by assessing the error norm. However, the efficiency of obtaining the solution as well as the rate of convergence of the iterative strategy are also important factors for real-field application, and therefore topic of future investigation.



(a) (b)

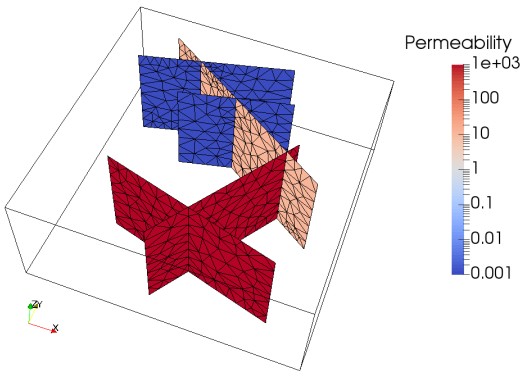


(c) (d)

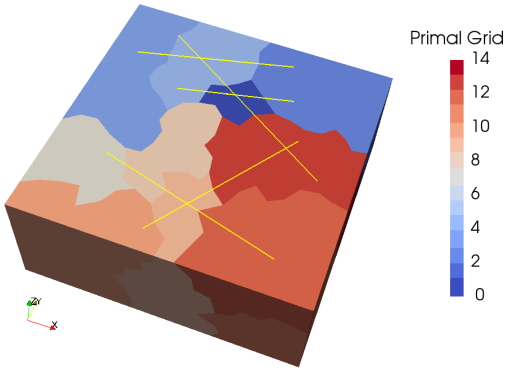
Figure 5.10: Comparison of fine reference solution and MS-DFM solutions (with no iterations) of different levels of refinement: (a) 5, (b) 12 and (c) 20 coarse matrix cells with 2, 8 and 18 coarse fracture cells, respectively .

5.4. MS-DFM: 3D test case

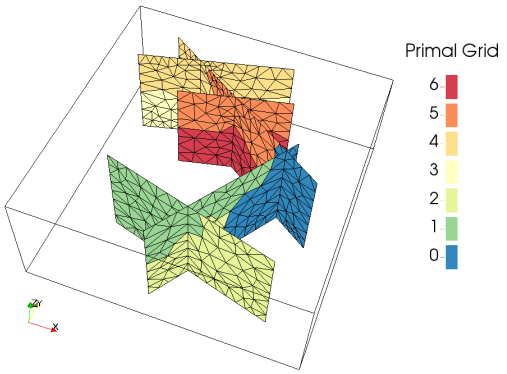
Unstructured 3-dimensional domains prove notably more challenging than lower dimensional ones. Therefore, MS-DFM's performance is assessed on a 3D test case, $1 \times 1 \times 0.4$, with 5 heterogeneous fractures. Two of the fracture plates are sealing, two are highly conductive and the last one has a slightly increased permeability with respect to the matrix, i.e, $k_{frac} = 10k_m$. This set-up is illustrated in Fig. 5.11a. The matrix is homogeneous and is assigned $k_m = 1$. The domain is gridded with 10025 matrix cells and 884 discrete fracture cells. The coarse grid consists of 15 matrix and 7 fracture cells. The coarse grids of both domains and the corresponding fracture dual-grid are depicted in Figures 5.11b, 5.11c and 5.11d respectively. The field is subjected to a line drive of 4 injector wells spaced equidistantly along the y-axis at $[x = 0.05, z = 0.35]$ and 4 low producer wells placed at mirrored locations along $[x = 0.95, z = 0.05]$.



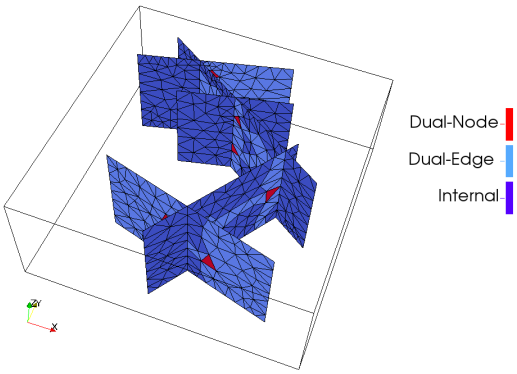
(a) Fracture Permeability



(b) Matrix Coarse Grid



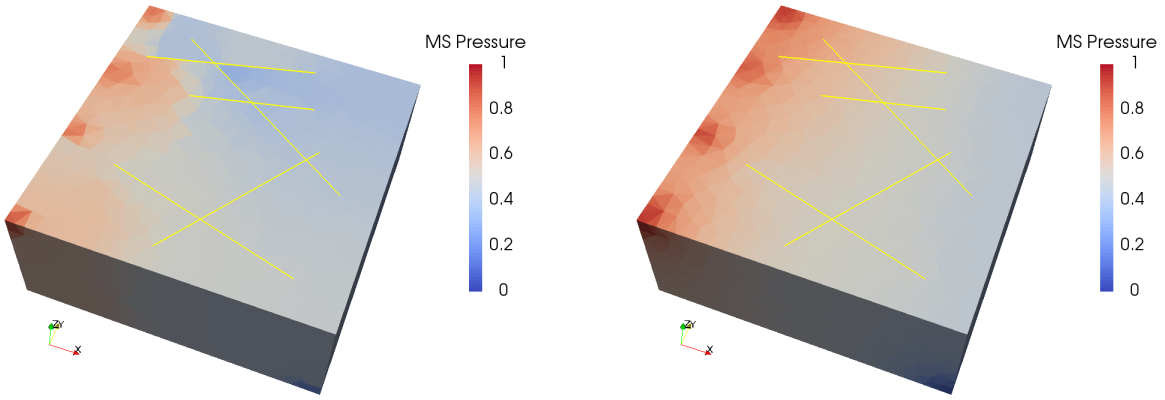
(c) Fracture Coarse Grid



(d) Fracture Coarse Dual-grid

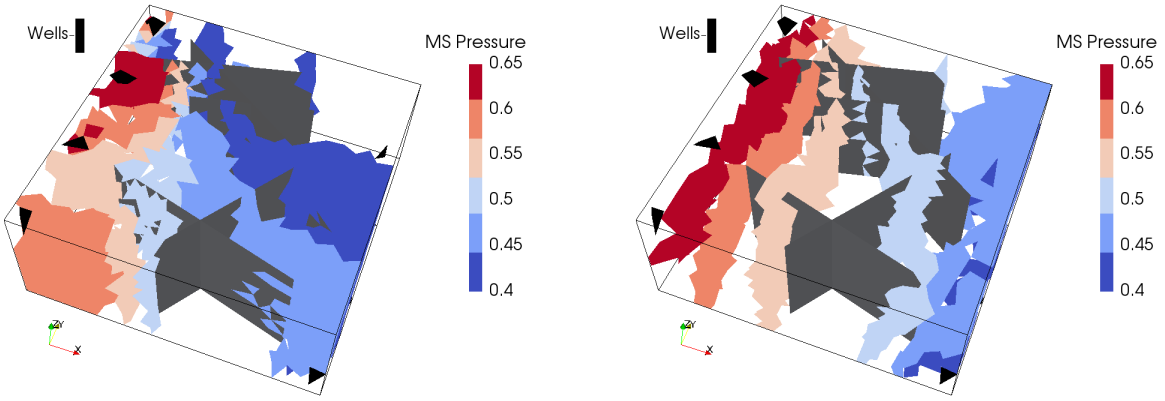
Figure 5.11: 3D test case permeability map, and coarse grids for matrix and fracture domain. Note that the matrix is homogeneous and is assigned non-dimensional permeability $k = 1$.

The MS-DFM result without iterations and converged reference solution are depicted in Fig. 5.12. Here, the MS-DFM approximation without iterations fails to correctly capture the effect of the sealing fractures in first instance. However, applying the iterative solving strategy, MS-DFM rapidly converges to the fine scale reference. By means of isobar plots, figures 5.13a to 5.13c illustrate the MS-DFM solution after 0, 2 and 10 iterations, respectively. After only 10 iterations, MS-DFM already closely approximates the converged reference presented in Fig. 5.13d.



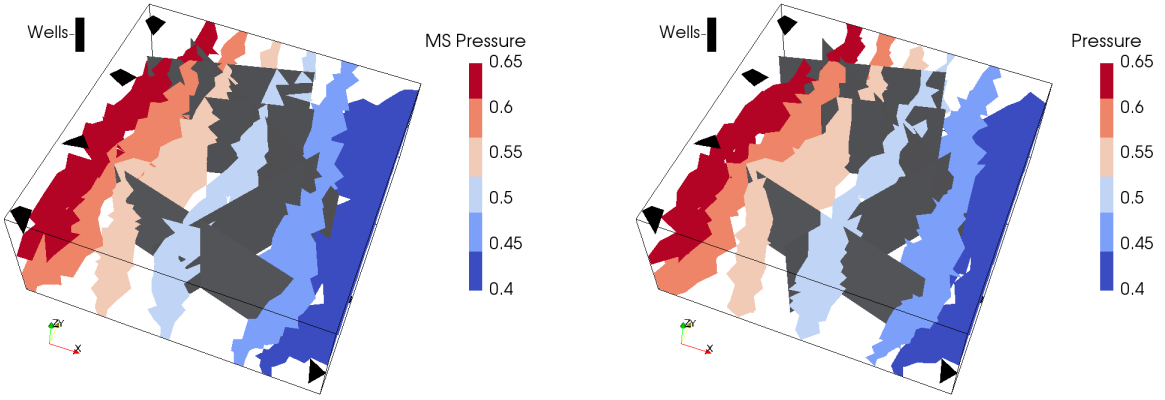
(a) MS-DFM (no iterations) (b) Reference

Figure 5.12: Comparison between MS DFM approximation with no iterations and the converged reference solution.



(a) MS-DFM 0 iterations
$|r|^2 = 2.43 \times 10^{-1}$

(b) MS-DFM 2 iterations
$|r|^2 = 5.64 \times 10^{-3}$



(c) MS-DFM 10 iterations
$|r|^2 = 7.12 \times 10^{-4}$

(d) Reference

Figure 5.13: Isobar plots illustrating convergence of the MS DFM solution of the 3D test case. Solutions after 0, 2 and 10 iterations are compared with the converged reference solution.

6

Conclusion

In this work, the first multiscale finite volume method for discrete fracture modeling on unstructured grids was developed and presented for 2D and 3D cases. The novelty of the development is two-fold: (1) the first MSFV method for fully unstructured 3-dimensional grids and (2) the first for discrete fracture modeling. Further extension with an iterative solving strategy was also presented, thus allowing for mass-conservative convergence to the reference solution. The proposed MS-DFM enables large flexibility in input fracture and matrix grids while at the same time only requiring geometrical data. Hence, the method allows for a straightforward black box integration within existing advanced simulators. Moreover, the method follows similar principles to the MSFV method on Cartesian grids and therefore permits the integration of proven extensions. This is subject of future research. Furthermore, the method is proven to be accurate for 2D and 3D test cases, specially with no iterations. Although currently allowing for any coarse partitioning, we point out the importance of the coarse grid geometries. As such, optimization of the partitioning and construction of the dual-grid is topic of future investigation. To conclude, the proposed MS-DFM framework proves a promising solving strategy for real-field DFM applications.

6.1. Future Work

Proven to be a promising method for field-scale fractured reservoir simulation, the MS-DFM framework opens the door to a whole new domain of research. In this subsection, topics for further research are pointed out in the direction of method optimization, efficiency and the extension to complex physics beyond the field of reservoir simulation.

6.1.1. Optimization

With the main objective of this work being method development, MS-DFM can still be further optimized to increase efficiency and accuracy. First of all, the algorithm to construct dual-faces was found to be relatively computationally expensive. Investigation into alternative strategies could notably decrease CPU time. On the accuracy side, optimization of the coarse grid geometry with respect to wells and fractures is of importance. Following the motivation mentioned in section 5.3.1, gridding strategies proved inevitably linked to the accuracy of the multiscale approximation without iterations. Another important topic related to gridding is the optimization of coarsening ratios, $n_{finecells}/n_{coarsecells}$. Combined with a consistent coarse grid strategy, an optimal coarsening ratio would enhance the black-box applicability of the method. Finally, accuracy of the multiscale approximation can be improved through a two-way coupling strategy between matrix and fractures when solving for basisfunctions. Although this imposes additional strain on the dual-grid construction, further research is necessary to perform a reasoned trade-off.

6.1.2. Efficiency

As an important incentive for the implementation of multiscale, further investigation into efficiency enhancing strategies is naturally of interest. Current multiscale extensions on Cartesian grids, such as parallel processing and multi-level multiscale, could prove equally beneficial for unstructured grids. However implementation of both strategies becomes notably more complex in the unstructured domain due to overlapping edges and complicated coarse grids. Ultimately, a CPU time comparison with industry standard linear solvers is necessary to appropriately benchmark MS-DFM.

6.1.3. Complex Physics & Beyond Reservoir Engineering

A logical next step in the development of MS-DFM is the extension to complex physics. Most notably physics benefiting from, or requiring, unstructured grids can now successfully be implemented in a conservative multiscale framework. The extension to incorporate geomechanics, for example, is of significant interest. Naturally, extensions to flow-related physics as well as proven extensions for multiscale methods on Cartesian grids are also subject of future research.

The presented method is of course not restricted to the described problem but could be extended to different physics. To this end, unstructured MS and MS-DFM could be generalized to the form of algebraic solvers based on connectivities.

A

TPFA formulation comparison

Although a widely implemented method, general consensus about the formulation of the classical two-point flux approximation (TPFA) is still non-existent. As the various formulations provide notably differing flux solutions on unstructured grids, a study is conducted into the most accurate approximation for this work's purpose.

The two most applied TPFA formulations are assessed. Written for half-transmissibility values, the equations are described as follows:

$$\text{(TPFA 1)} \quad T_{if} = \frac{A_f * k_i}{|\vec{v}_{cf}|^2} \vec{n}_f \cdot \vec{v}_{cf} \quad [9][14], \tag{A.1}$$

and

$$\text{(TPFA 2)} \quad T_{if} = \frac{A_f * k_i}{\vec{n}_f \cdot \vec{v}_{cf}} \quad [28]. \tag{A.2}$$

Here, A_f is the face area, k_i is the scalar value of the cell permeability, $\vec{n}_f$ is the unit normal vector to the face, and $\vec{v}_{cf}$ is the vector from cell center to face center.

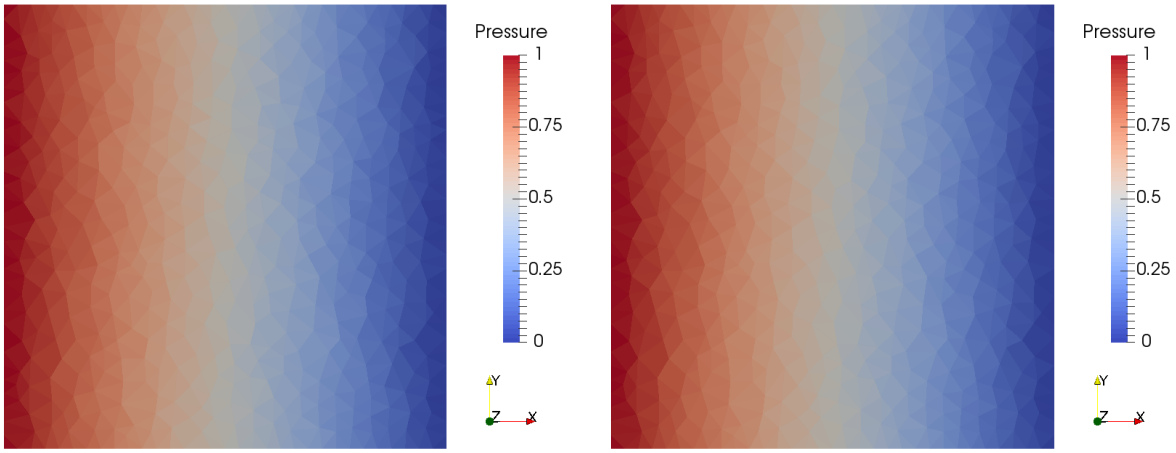
To quantify the performance of the two methods, a simple homogeneous test case in a $[1 \times 1 \times 0.5]$ domain is devised. The grid contains 19882 cells and is subject to a non-dimensional Dirichlet boundary condition of $p = 1$ at $x = 0$ and $p = 0$ at $x = 1$. The results are displayed in Fig. A.1.

The solutions do not show any discernible differences when examining the plots. However when computing the outgoing flux over the domain face at $x = 1$, the accuracy of the methods varies greatly. Following Darcy's law, the analytical flux over the face is expected to be

$$q = A\lambda \frac{\Delta p}{L} = 0.5 * 1 * \frac{1}{1} = 0.5, \tag{A.3}$$

where A is the face area, λ the mobility, Δp the pressure drop and L the domain length. To find the outgoing flux in the numerical solutions, the flux over all the cell faces at the specified domain boundary are summed. This results in a value of $q = 0.3670$ for TPFA 1 and $q = 0.5299$ for TPFA 2. As such we can conclude that TPFA 2 is significantly more accurate for our purpose and, thus, will be applied throughout this research.

Note that in the case that k_i is a tensor, it can be projected on either $\vec{n}_f$ or on the unit vector of $\vec{v}_{cf}$. As only scalar values for permeability are used in this work, further investigation into the different projections is not relevant and therefore subject of future research. However, preference is given to projection on $\vec{n}_f$. This due to the fact that Gauss' theorem defines flux direction as normal to the cell boundary. Additionally, matrix cell interaction is not forced through a common interface because flow through an intermediate cell is physically not restricted in the targeted applications. Therefore, we obtain the formulation as presented in Eq. (2.4).



(a) TPFA 1

(b) TPFA 2

Figure A.1: Top view of 3D test case for comparison of TPFA methods

B

Verification pEDFM: Complex 3D Test Case

The fine scale DFM simulator was also implemented to verify the newly developed Projection-based Embedded Discrete Fracture Model [27]. pEDFM provides a significantly improved alternative to the classical EDFM approach where the new model accurately captures the effects of both highly permeable fractures and flow barriers. The method is described and extensively proven in [27]. By adapting the fracture-matrix connectivity through the projection of fractures onto matrix cell faces, pEDFM adds additional connections to fractured matrix cells, their neighbors and fracture cells. Through this projection strategy, the method also reduces to the traditional discrete fracture model when fractures are at matrix interfaces. In the cited work, to show pEDFM's general validity, a complex 3D test case with 3 heterogeneous fracture sets is devised. The permeability and geometry of the fracture sets are depicted in Fig. B.1. Non-dimensional Dirichlet boundary conditions of $p = 1$ and $p = 0$ are imposed on the left- and right-hand side, respectively. The structured grid with pEDFM consisted of 10^6 matrix cells and 23381 fracture cells. The unstructured grid for DFM has 999933 matrix cells and 32978 fracture cells. Note that these dimensions are matched as accurately as possible considering the unstructured gridding restrictions.

The pressure results are depicted by means of isobar plots in Fig. B.2 and the solutions are in good agreement.

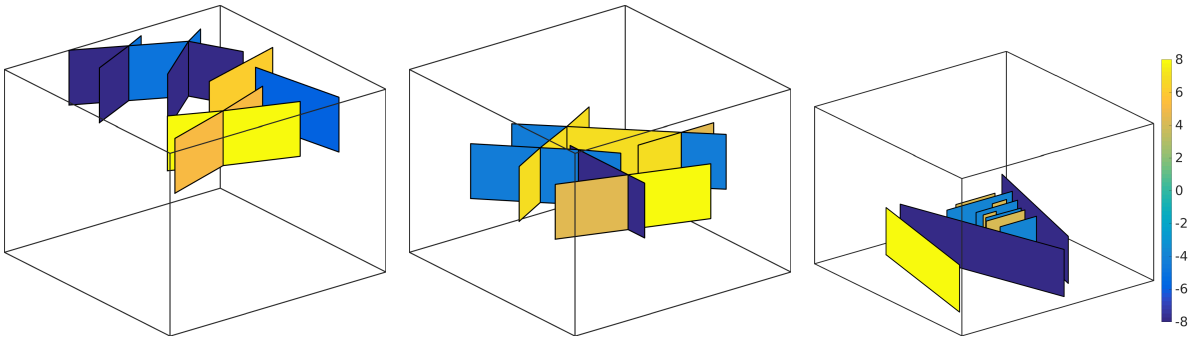
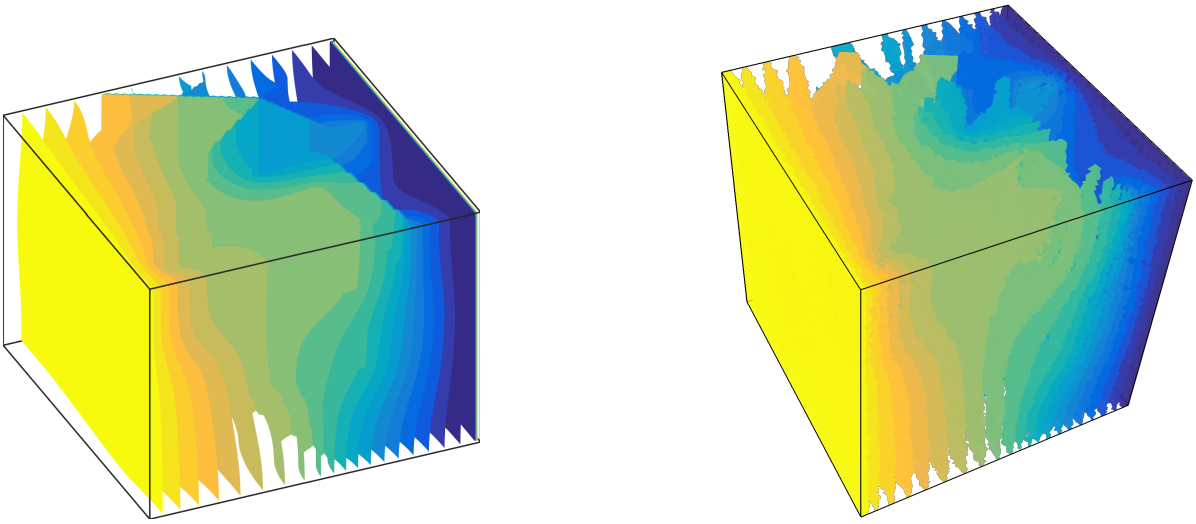


Figure B.1: Illustration of the complex fracture networks and their permeability in log scale.



(a) p-EDFM (b) DFM

Figure B.2: Comparison of DFM and pEDFM with isobar plots of complex 3D test case.

Bibliography

[1] R Ahmed, Michael G Edwards, Sadok Lamine, B A H Huisman, and M Pal. Control-volume distributed multi-point flux approximation coupled with a lower-dimensional fracture model. *J. Comput. Phys.*, 284:462–489, 2015.

[2] M. Cusini, C. van Kruijsdijk, and H. Hajibeygi. Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media. *J. Comput. Phys.*, 314:60–79, 2016.

[3] Y. Efendiev and T. Y. Hou. *Multiscale Finite Element Methods: Theory and Applications*. Springer, 2009.

[4] H. Hajibeygi and H. A. Tchelepi. Compositional multiscale finite-volume formulation. *SPE J.*, 19:316–326, 2014.

[5] H. Hajibeygi, G. Bonfigli, M.A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *J. Comput. Phys.*, 227:8604–8621, 2008.

[6] H. Hajibeygi, D. Karvounis, and P. Jenny. A hierarchical fracture model for the iterative multiscale finite volume method. *J. Comput. Phys.*, 230:8729–8743, 2011.

[7] T. Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[8] P. Jenny, S. H. Lee, and H. A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.*, 187:47–67, 2003.

[9] M. Karimi-Fard, L.J. Durlofsky, and K. Aziz. An efficient discrete-fracture model applicable for general-purpose reservoir simulators. *SPE J.*, 9:227–236, 2004.

[10] G. Karypis. Metis - serial graph partitioning and fill-reducing matrix ordering. Technical Report version 5.1.0, Karypis Lab, University of Minnesota, 2013.

[11] S. H. Lee, C. Wolfsteiner, and H. A. Tchelepi. Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. *Comput. Geosci.*, 12:351–366, 2008.

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

[13] A. Moinfar, A. Varavei, K. Sepehrnoori, and R. T. Johns. Development of an efficient embedded discrete fracture model for 3d compositional reservoir simulation in fractured reservoirs. *SPE J.*, 19:289–303, 2014.

[14] G. Moog. *Advanced Discretization Methods for Flow Simulation using Unstructured Grids*. Phd thesis, Stanford University, United States, 2013.

[15] O. Moyner and K.A. Lie. A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. *J. Comput. Phys.*, 304:46–71, 2016.

[16] M.Tene, Mohammad Saad Al Kobaisi, and H. Hajibeygi. Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (f-ams). *J. Comput. Phys.*, 321:819–845, 2016.

[17] E. Parramore, G. Edwards, M. Pal, and S. Lamine. Multiscale finite-volume cvd-mpfa formulations on structured and unstructured grids. *SIAM*, 14, no2:559–594, 2016.

[18] D. Peaceman and H. Rachford. The numerical solution of elliptic and parabolic differential equations. *Journal of SIAM*, 3:28–41, 1955.

[19] S. Pluimers. *Hierarchical fracture modeling*. Msc thesis, Delft University of Technology, The Netherlands, 2015.

[20] V. Reichenberger, H. Jakobs, P. Bastian, and Rainer Helmig. A mixed-dimensional finite volume method for two-phase flow in fractured porous media. *Advances in Water Resources*, 29:1020–1036, 2006.

[21] Schlumberger. Characterization of fractured reservoirs. 2008.

[22] S. Shah, O. Moyner, M.Tene, K. Lie, and H. Hajibeygi. The multiscale restriction smoothed basis method for fractures porous media (f-msrsb). *J. Comput. Phys.*, 318C:36–57, 2016.

[23] J. Shewchuk. Triangle - a two-dimensional quality mesh generator and delaunay triangulator. Technical Report version 1.6, University of California at Berkeley, 2005.

[24] H. Si. Tetgen - a quality tetrahedral mesh generator and a 3d delauney triangulator. Technical Report version 1.4.3, Weierstrass Institute for Applied Analysis and Stochastics, 2011.

[25] M. Tene, Y. Wang, and H. Hajibeygi. Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media. *J. Comput. Phys.*, 300:679–694, 2015.

[26] M. Tene, M.S. Al Kobaisi, and H. Hajibeygi. Multiscale projection-based embedded discrete fracture modeling approach (f-ams-pedfm). *Proceedings of the 15th European Conference on the Mathematics of Oil Recovery, ECMOR XV*, 2016. doi: 10.3997/2214-4609.201601890.

[27] M. Tene, S.B.M. Bosma, M.S. Al Kobaisi, and H. Hajibeygi. Projection-based embedded discrete fracture model (pedfm). *Advances in Water Resources*, Submitted 2017.

[28] K. Terekhov and Yu. Vassilevski. Two-phase water flooding simulations on dynamic adaptive octree grids with two-point nonlinear fluxes. *Russian J. of Numerical Analysis and Mathematical Modelling*, 28(3):267–288, 2013.

[29] John Wallis and Hamdi A Tchelepi. Apparatus, method and system for improved reservoir simulation using an algebraic cascading class linear solver, March 2010. US Patent 7,684,967.

[30] Y. Wang, H. Hajibeygi, and H. A. Tchelepi. Algebraic multiscale linear solver for heterogeneous elliptic problems. *J. Comput. Phys.*, 259:284–303, 2014.

[31] C. Wolfsteiner, S.H. Lee, and H.A.Tchelepi. Well modeling in the multiscale finite volume method for subsurface flow simulation. *Multiscale Model. Simul.*, 5(3):900–917, 2006.

[32] H. Zhou and H. A. Tchelepi. Operator based multiscale method for compressible flow. *SPE J.*, 13:267–273, 2008.

[33] H. Zhou and H. A. Tchelepi. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. *SPE J.*, 17(2):523–539, 2012.

[34] H. Zhou, S.H. Lee, and H.A. Tchelepi. Multiscale finite-volume formulation for saturation equations. *SPE J.*, 17:198–211, 2011.