

Multiscale finite volume method for discrete fracture modeling on unstructured grids (MS-DFM)



Sebastian Bosma ^{a,*}, Hadi Hajibeygi ^a, Matei Tene ^a, Hamdi A. Tchelepi ^b

^a Department of Geoscience and Engineering, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands

^b Department of Energy Resources Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:

Received 20 March 2017

Received in revised form 15 September 2017

Accepted 18 September 2017

Available online 21 September 2017

Keywords:

Multiscale finite volume

Unstructured grids

Discrete fracture modeling

Flow in fractured porous media

Algebraic multiscale

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. The proposed method achieves accurate results for several test cases even without iterations, and for challenging ones with only a few 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.

© 2017 Elsevier Inc. All rights reserved.

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 [1]. 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 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

* Corresponding author.

E-mail addresses: sbosma@stanford.edu (S. Bosma), H.Hajibeygi@tudelft.nl (H. Hajibeygi), M.Tene@tudelft.nl (M. Tene), Tchelepi@stanford.edu (H.A. Tchelepi).

embedded discrete fracture modeling (EDFM) approach has been developed [2–5]. More recently, the projection-based EDFM method enables the simulation of a wider range of conductivity contrasts including flow barriers [6,7]. Alternatively, a different approach uses unstructured grids to place fractures at the interface between matrix cells [8–11]. 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 [12]. However, 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. Recently, a comparison benchmark study between several discrete fracture modeling approach has been presented [13]. 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 [14–17]. 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 [18–21]. 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 [15,22–25].

Despite multiple extensions of MS methods, the majority have been applied to structured Cartesian grids. Additionally, the limited amount of investigations for unstructured grids have either restricted coarse grid flexibility or restricted fine grid cell geometries [26–28]. Note that the literature includes only limited investigations of either 2-Dimensional (2D) test cases or simplified 3D cases obtained by extruding the 2D geometries in the third dimension (referred to as 2.5D).

The first MSFV method for 2D fractured porous media was developed based on single fracture basis functions per network in order to add one degree-of-freedom (DOF) per network in the coarse system [3]. This method proved efficient for media with highly conductive fracture networks which span relatively short spatial length scales. However, convergence was observed to degrade for test cases with significant variations in the pressure distribution along the fracture network. Later, an alternative 2D approach was developed which applies the fracture networks as the dual-coarse grid, with dual-nodes at the intersections [29]. This approach was found effective only for cases with highly conductive fractures and a high density of fracture networks. Unfortunately, this method did not allow for a flexible coarse grids, neither did it allow for independent coarse grids (for matrix and fractures). Using streamline-based mixed formulations, multiscale methods have also been employed to 2D fractured reservoirs [30]. Recently, a general formulation for the EDFM approach was developed on structured [31] and unstructured 2.5D grids [32]. Note that the latter reference avoids dual grids to enable global iterative construction of basis functions. In summary, although promising progress has been achieved, the challenge of combining a fully flexible 3D multiscale method on unstructured 3D grids for discrete fracture modeling has not yet been resolved.

In this work, a multiscale method for the Discrete Fracture Modeling approach on fully unstructured 3D grids (MS-DFM) is devised. To this end, we extend the state-of-the-art MS methods to include the DFM formulation with complex grids. In comparison to existing MS methods for fractures [31,32,29], MS-DFM requires a more intelligent construction of coarse grids guided by geometrical rules. The resulting complex coarse grids naturally asks for a new treatment when solving for basis functions to ensure strictly local solutions while preserving the partition of unity. Additionally, a coupling strategy suited for these complex coarse grids is necessary. As a result MS-DFM allows for full flexibility in coarse grid geometry, naturally treats flow barriers accurately, implements a consistent coupling within and between the fracture and matrix domains, and can handle flexible input grid geometries.

To overcome the aforementioned hurdles, a general procedure is devised. Starting from 2 independent arbitrary partitions, i.e., one for the matrix and one for the fractures, a 3D wire-basket ordering [33,34] 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.

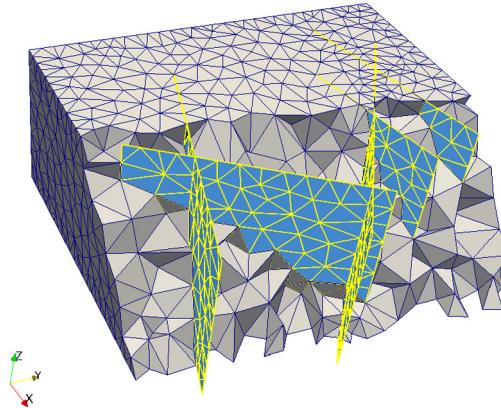


Fig. 1. View of conforming 3D grid cells.

The paper 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. Subsequently, systematic numerical tests are presented to first validate the fine-scale model, and then assess the performance of 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

$$-\nabla \cdot (\lambda \cdot \nabla p) = q, \quad (1)$$

where λ , p , and q are the mobility, pressure, and source terms, respectively. Equation (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 [8].

The matrix grid is generated such that fracture elements are confined to the matrix cell interfaces. Fig. 1 illustrates an example of such a 3D domain. In this work, the mesh generators TetGen [35] and Triangle [36] are used for 3-dimensional (3D) and 2-dimensional (2D) domains, respectively.

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 neighboring cells can be expressed in a generic form, irrespective to which domain (fracture or matrix) they belong. Applying a finite-volume method, Eq. (1) is discretized for every control volume, i.e., matrix and fracture. In the classical two-point flux approximation scheme [37], implemented in this work, the discrete system reads

$$\text{for every cell } i: \sum_{j=1}^{n_{nb}} T_{ij}(p_j - p_i) = q_i, \quad (2)$$

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

$$T_{ij} = \frac{T_{if} T_{jf}}{T_{if} + T_{jf}}, \quad (3)$$

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

$$T_{if} = \frac{A_f * (\vec{n}_f \cdot (\mathbf{k}_i \cdot \vec{n}_f))}{\vec{n}_f \cdot \vec{v}_{cf}}. \quad (4)$$

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 . Fig. 2 illustrates these variables for a 2D cell (hence, the area is replaced by the interface length L_f).

To avoid small control volumes at fracture intersections, the star-delta procedure is applied [8]. In this case, the sum of all half-transmissibilities acting on the intersection is the denominator of Eq. (3).

Following the described procedure, the linear system corresponding to Eq. (2) finally reads

$$A p \equiv \begin{bmatrix} A_{mm} & A_{mf} \\ A_{fm} & A_{ff} \end{bmatrix} \begin{bmatrix} p_m \\ p_f \end{bmatrix} = \begin{bmatrix} q_m \\ q_f \end{bmatrix} \equiv q, \quad (5)$$

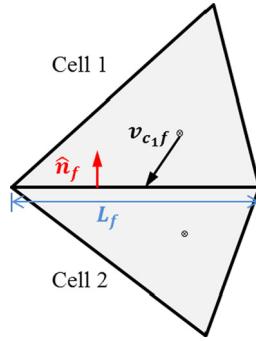


Fig. 2. Half transmissibility variables in a 2D cell.

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 (5) provides an accurate (fine-scale) solution to Eq. (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. (5).

3. MSFV on unstructured grids

In this section, the multiscale finite volume method (MSFV) [15] 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 [26,32]. 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 [20]. 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. (5), based on a superposition expression which can be algebraically stated as

$$p \approx p' = \mathbf{P} \check{p}. \quad (6)$$

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 $\check{\mathbf{A}}\check{p} = \check{q}$, which is constructed algebraically as

$$\underbrace{(\mathbf{R} \mathbf{A} \mathbf{P})}_{\check{\mathbf{A}}} \check{p} = \underbrace{\mathbf{R} q}_{\check{q}}. \quad (7)$$

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

$$R(i, j) = \begin{cases} dv_j & \text{if fine-cell } j \text{ belongs to coarse cell } i \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

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 [20,21].

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

```

1:  $v = 1$ 
2:  $p^v = 0$            Initiate
3:  $r^v = q - Ap^v$    Compute residual
4:
5: while ( $\|r\|_2^v > \epsilon$ ) do
6:   Multiscale stage:
7:    $\check{r}^{v+1/2} = \mathbf{R} r^v$            Compute coarse residual
8:    $\delta\check{p}^{v+1/2} = \tilde{\mathbf{A}}^{-1} \check{r}^{v+1/2}$    Compute coarse pressure correction
9:    $\delta p^{v+1/2} = \mathbf{P} \delta\check{p}^{v+1/2}$    Prolong to find fine pressure correction
10:   $p^{v+1/2} = p^v + \delta p^{v+1/2}$    Update pressure
11:   $r^{v+1/2} = q - Ap^{v+1/2}$    Update residual
12:
13:  Smoother stage:
14:   $p^{v+1} = p^{v+1/2} + \mathbf{M}_{ILU(0)}^{-1} r^{v+1/2}$    Apply Smoother
15:   $r^{v+1} = q - Ap^{v+1}$    Update residual
16:
17:   $v \leftarrow v + 1$ 
18: end while

```

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

$$\mathbf{P} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \Phi_1^m & \dots & \Phi_i^m & \dots & \Phi_{N_{mc}}^m \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}_{N_f \times N_c}. \quad (9)$$

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 [18,31]. 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 [Algorithm 1](#).

3.3. Primal and dual grid

To construct the restriction and prolongation operators, a primal and dual coarse grid are required. The primal grid represents the coarse grid, or partitioning, and is constructed using METIS [38] 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. [Figs. 3 and 4](#) depict a primal- and its corresponding dual-coarse cell for a 2D test case.

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 [Algorithm 2](#).

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

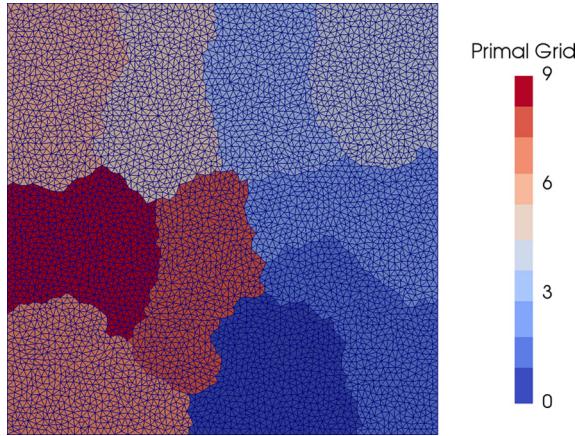


Fig. 3. Coarse grid imposed on a 2D unstructured fine grid, obtained with METIS [38]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

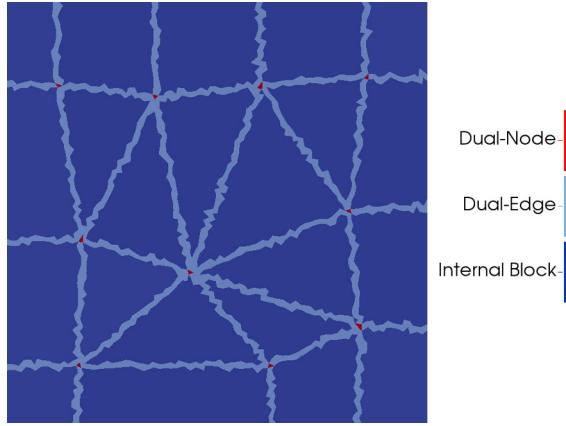


Fig. 4. Dual-Grid corresponding to the partitioning of Fig. 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2. Construction of dual-edges.

```

1: for  $i = 0$  to  $n_{dualnodes}$  do
2:   for all  $j = \text{connected dualnodes}$  do
3:     CurCell = IniCell
4:     while CurCell  $\neq$  TargetCell do
5:       Assign CurCell to the dual-edge
6:       Find intersected interface of CurCell
7:       CurCell  $\leftarrow$  neighbor of intersected interface
8:     end while
9:   end for
10: end for

```

▷ IniCell is the fine cell of dual-node i
▷ TargetCell is the fine cell of dual-node j

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.,

$$\text{Code} = \begin{cases} 1 & \text{if } (\vec{n}_{fp} \cdot \vec{v}) > 0 \\ 2 & \text{otherwise,} \end{cases} \quad (10)$$

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 [Algorithm 3](#), where CellNodeCodes is the set of codes present in the grid nodes constructing the fine cell.

Algorithm 3. Construction of a dual face.

```

1: for  $i = 0$  to  $n_{gridnodes}$  do
2:   Project node on Face Plane
3:   if Projection inside virtual face (check with barycentric coordinates) then
4:     if Node is above face plane (see Eq. (10)) then
5:       Node assigned code 1
6:     else
7:       Node assigned code 2
8:     end if
9:   else if Projection outside virtual face then
10:    Node assigned code 0
11:   end if
12: end for
13:
14: for all  $i = UnassignedCells$  do
15:   if  $\{1, 2\} \in CellNodeCodes$  then
16:     Cell i assigned to the dual-face
17:   end if
18: end for

```

Algorithm 4. Construction of internal blocks.

```

1: for  $i = UnassignedCells$  do
2:   Initiate new block
3:   Assign cell i to block
4:   Initiate list CurrentCells with i
5:   while  $n_{CurrentCells} > 0$  do
6:
7:     for all  $j = CurrentCells$  do
8:       for all  $k = Unassigned$  neighbors of  $j$  do
9:         Assign cell k to block
10:        Add cell k to NewCells
11:      end for
12:    end for
13:
14:    CurrentCells  $\leftarrow$  NewCells
15:
16:   end while
17: end for

```

▷ Thus cell k removed from $UnassignedCells$

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 Section 3.4), cells are assigned a unique code per block. To do so, a flood-fill algorithm is implemented (Algorithm 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.

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 Section 3.4). The edge networks of the previously displayed 2D test case are shown in Fig. 5.

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. Nevertheless, efficiency improvements will be investigated in future work.

3.4. Basis functions

Physics-based basis functions are obtained by solving the governing equation, Eq. (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

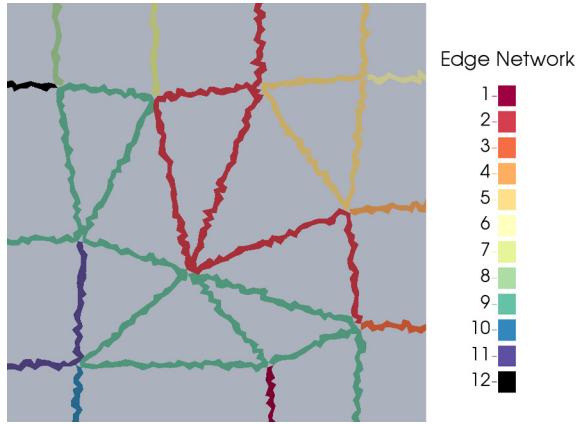


Fig. 5. Edge networks in 2D example of Fig. 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 5. Basis function solving procedure.

```

1: for  $i = 0$  to  $n_{edgenetworks}$  do
2:   for all  $j =$  influencing dual-node do
3:     Set value Cell  $j = 1$ 
4:     Set value other influencing dual-nodes = 0
5:   Solve basis function for edge network
6: end for
7: Rescale solution for edge cells in edge-network  $i$  (Eq. (11))
8: end for
9:
10: for  $i = 0$  to  $n_{facenetworks}$  do
11:   for all  $j =$  influencing dual-node do
12:     Set neighboring dual-edge and dual-node cells to  $\phi_j$  value
13:   Solve basis function for face network
14: end for
15: Rescale solution for face cells in face-network  $i$  (Eq. (11))
16: end for
17:
18: for  $i = 0$  to  $n_{internalblocks}$  do
19:   for all  $j =$  influencing dual-node do
20:     Set neighboring dual-face, dual-edge and dual-node cells to  $\phi_j$  value
21:   Solve basis function for internal block
22: end for
23: Rescale solution for internal cells
24: end for
```

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 Algorithm 5. Fig. 6 depicts two basis functions resulting from the coarse grids displayed in Figs. 3 and 4. Here, the dual-node numbering follows the numbering of the coarse partitions, represented by each dual-node (see Fig. 3).

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.

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

$$\text{for cell } i: \mathbf{P}_{ij} = \begin{cases} \frac{\phi_{ij}}{\sum \phi_i} & \text{if basis function } j \in \mathbf{bf}_i \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

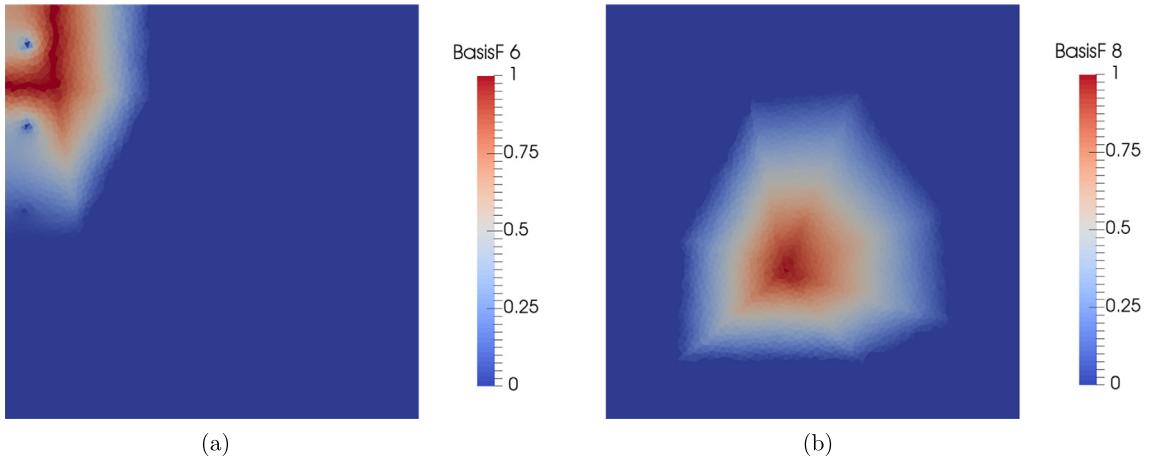


Fig. 6. Basis functions of dual-node 6 (a) and 8 (b). Note that basis function 6 is also influenced by 2 well basis functions [39]. Furthermore, the dual-node numbering follows from the numbering of the primal partitions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 \mathbf{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.

4. MSFV on unstructured grids with DFM

Fractures demand appropriate treatment within the multiscale framework [31]. 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 unstructured 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 Algorithm 2, line 6. The new criterion is evaluated 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 grid nodes of the assessed cell-edge are named A and C . Fig. 7 illustrates the vectors described. The code then evaluates the following set of conditions:

$$\text{if } ((A \cdot B < 0) \& (C \cdot B < 0)) \text{ then the edge is in the wrong direction} \quad (12)$$

and

$$\text{if } ((A \times B) \cdot (A \times C) < 0) \& ((C \times B) \cdot (C \times A) < 0) \text{ then } B \text{ intersects the edge.} \quad (13)$$

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.

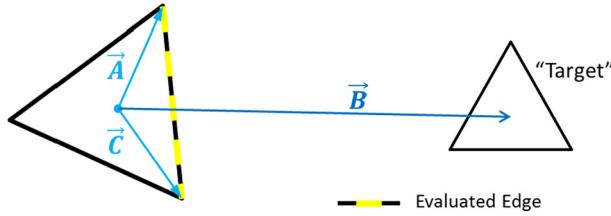


Fig. 7. Illustration of the vectors used when evaluating an edge.

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 for the 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 Section 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. (6), 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

$$p'^f = \sum_{i=1}^{N_{cf}} \Phi_i^f \check{p}_i^f, \quad (14)$$

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 [31] for more information), which leads to the

$$p'^m = \sum_{i=1}^{N_{cm}} \Phi_i^{mm} \check{p}_i^m + \sum_{i=1}^{N_{cf}} \Phi_i^{mf} \check{p}_i^f \quad (15)$$

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. (11)) still follows the same principles. The area of influence of a fracture basis function in the matrix domain is defined as all matrix structures directly connected to a fracture cell influenced by that fracture basis function. Fig. 8b shows such a fracture basis function in the matrix domain. Note that this is the same test case as in Figs. 3–6 where now the cell faces indicated by the yellow line are activated as highly conductive fractures. Fig. 8a shows the effect on the previously presented matrix basis function and Fig. 9 illustrates the total effect of all fracture basis functions in the matrix domain, i.e., $\sum \Phi^{mf}$.

4.3. Multiscale operators

The restriction operator \mathbf{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,

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^m \\ \mathbf{P}^f \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{mm} & \mathbf{P}^{mf} \\ 0 & \mathbf{P}^{ff} \end{bmatrix}, \quad (16)$$

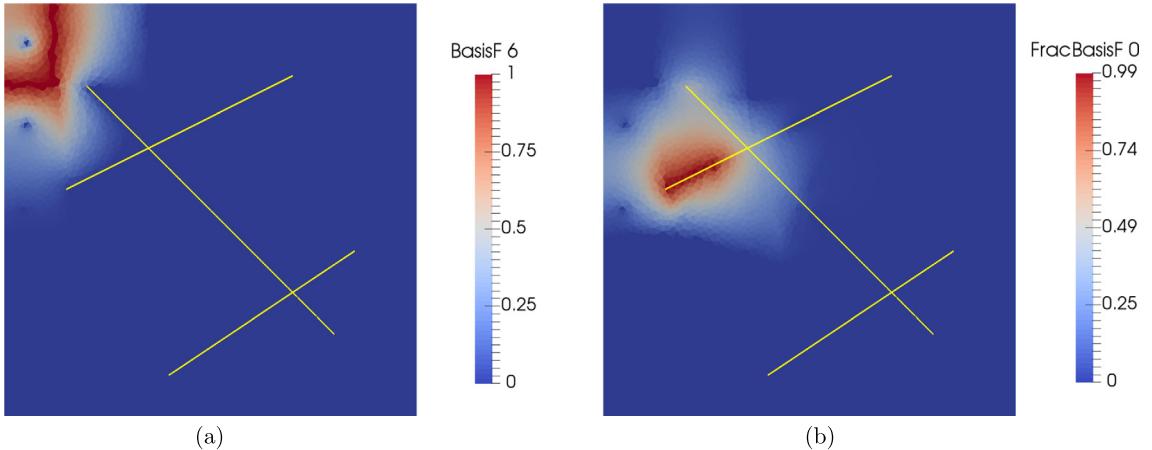


Fig. 8. (a) Basis function of a dual coarse node 6, i.e. Φ_6^{mm} , with fractures being set to a Dirichlet value of 0. (b) The basis function of fracture dual-node 0 in the matrix domain, i.e., Φ_0^{mf} . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

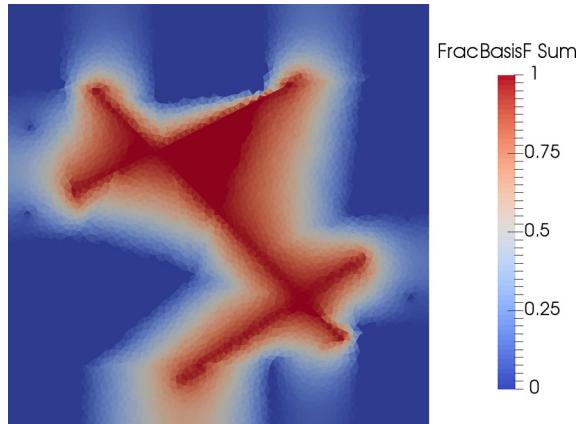


Fig. 9. Illustration of the effect of $\sum \Phi^{mf}$, i.e., the sum of all fracture basis functions in the matrix domain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where, more precisely,

$$\mathbf{P}^m = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \Phi_1^{mm} & \dots & \Phi_{N_{cm}}^{mm} & \Phi_1^{mf} & \dots & \Phi_{N_{cf}}^{mf} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \end{bmatrix}. \quad (17)$$

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.

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 [40]. 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

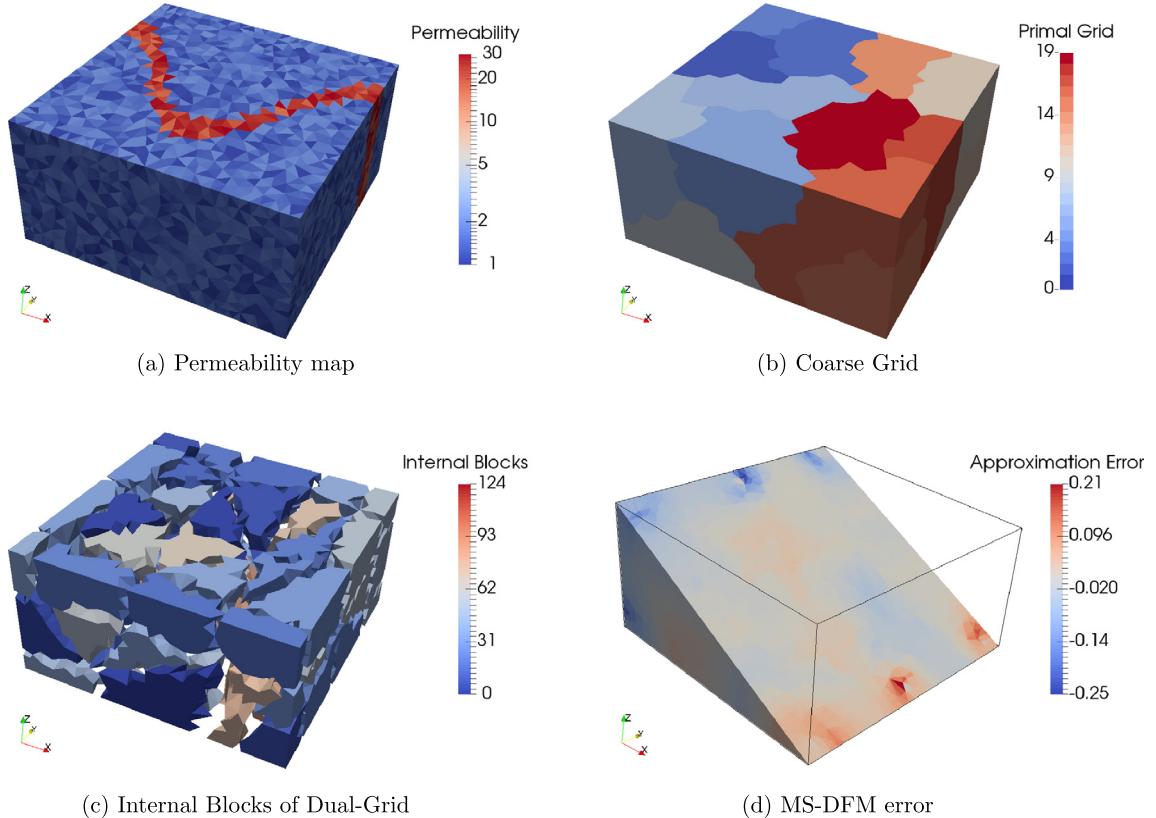


Fig. 10. Illustrations of the problem set-up, coarse grids and MS-DFM approximation error. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

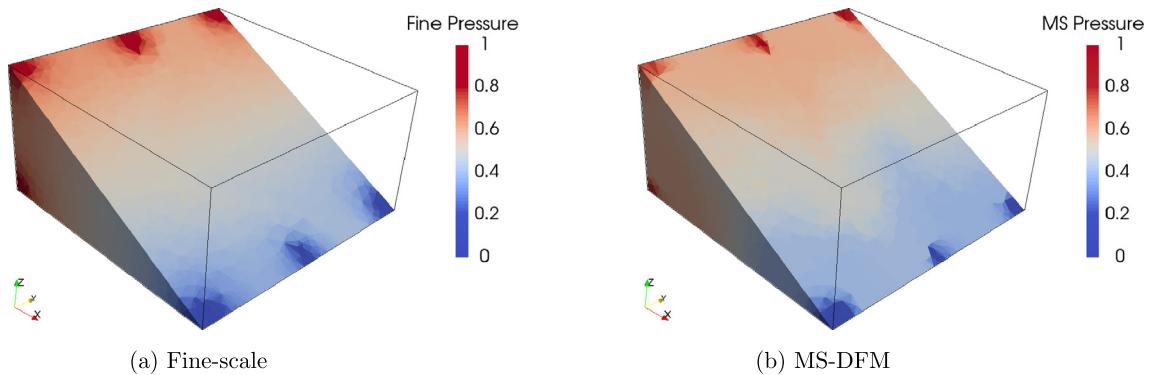


Fig. 11. Comparison of the fine-scale reference solution and MS-DFM solution (without iterations). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$z = 0.45$. 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. 10a. 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. 10b and 10c, respectively, where the dual grid is visualized by means of the internal blocks which are separated by the higher-ranked structures (dual-edges, dual-faces and dual-nodes).

The fine scale and multiscale solutions are depicted on a diagonal cross section of the matrix domain in Figs. 11a and 11b, 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. 10d. 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

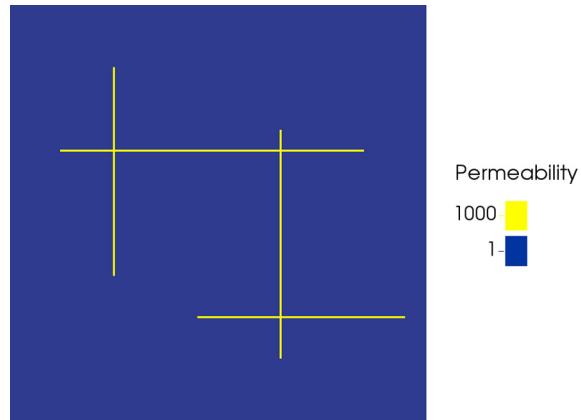
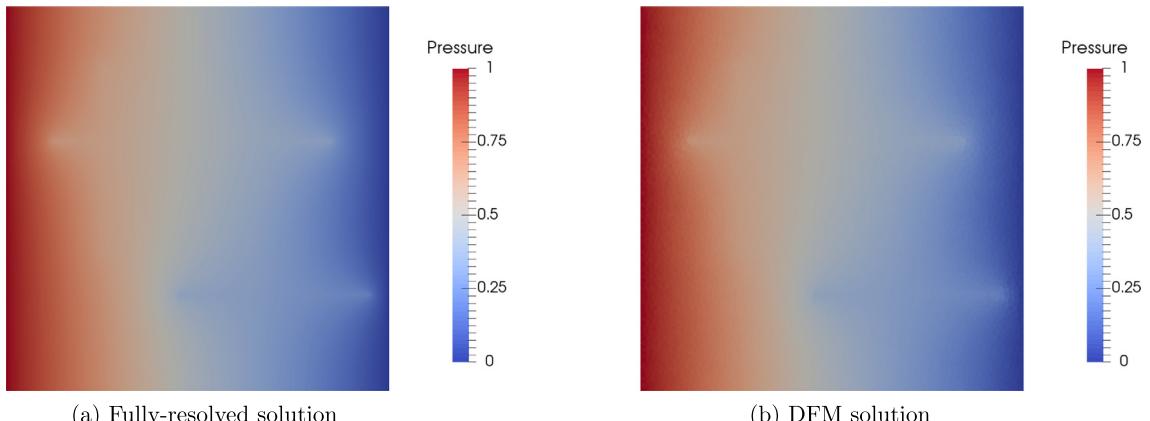


Fig. 12. The fracture-configuration used for the validation of the DFM model (fractures are emphasized for clarity). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Fully-resolved solution

(b) DFM solution

Fig. 13. Comparison of fully resolved and DFM solutions. Note that the fully resolved case employs 656100 cells, while the DFM employs 10003 cells. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

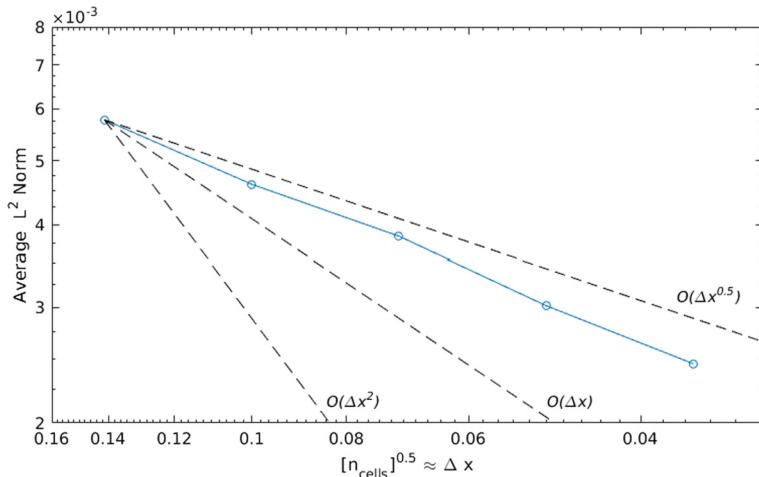


Fig. 14. Grid convergence of DFM (measured in L^2 norm at 100 fixed locations in the domain) with respect to the fully resolved simulation.

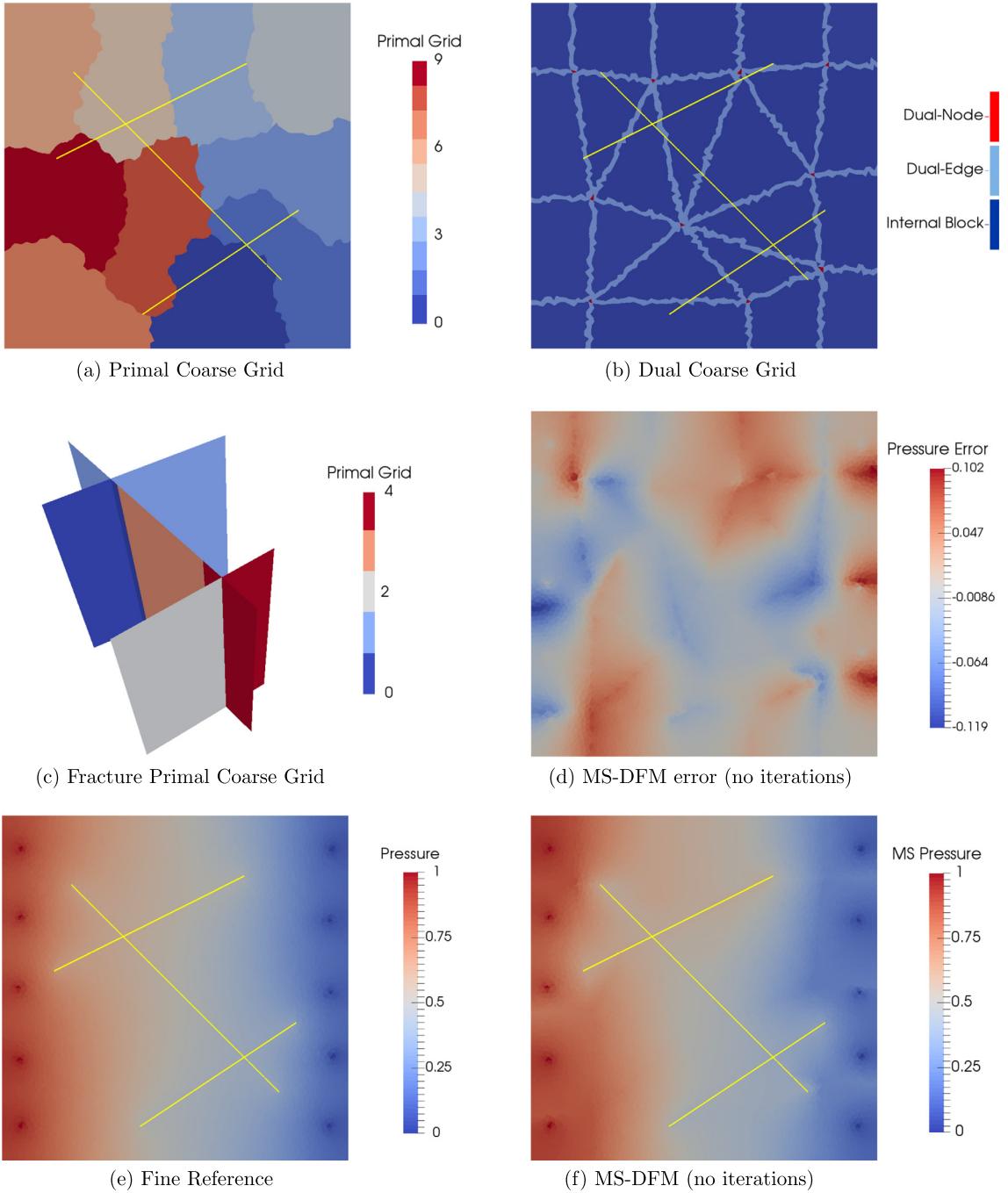


Fig. 15. Illustrations of the coarse primal grids for both fracture and matrix domain as well as matrix dual-grid are presented in (a)–(c). Subfigures (d)–(f) present the comparison of the fine reference and MS-DFM solution for the 2D Test Case. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

maximum discrepancy of 0.07 in the rest of the domain. Note that by applying iterations of the i-MSFV procedure, one can systematically reduce the error to any desired level.

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. 12, 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

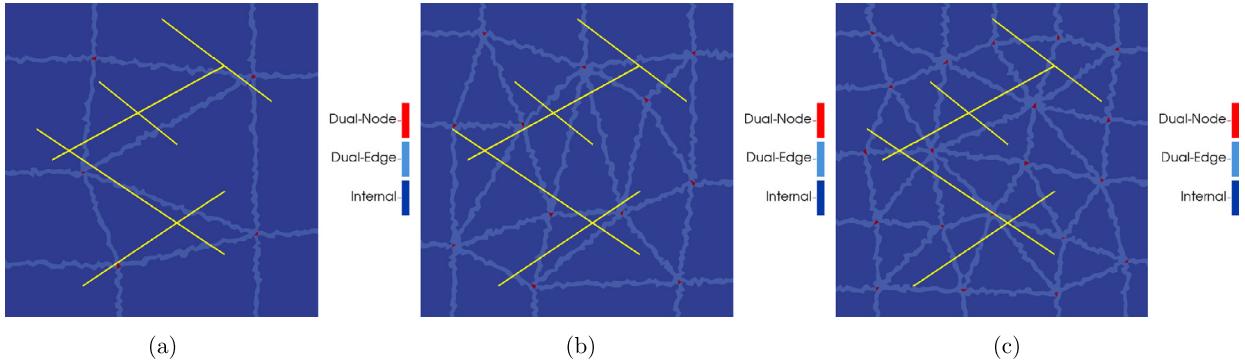


Fig. 16. Matrix coarse dual-grids for 3 levels of coarse grid refinement: (a) 5, (b) 12, and (c) 20 coarse matrix cells. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

resolving fractures with an 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. 13a and 13b. It is clear that the DFM model accurately captures the real solution. In more detail, Fig. 14 plots the average 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.

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 Section 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 [40]. 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. Figs. 15a to 15c present the primal grids of both the fracture and matrix domains as well as the matrix dual-grid.

Figs. 15e and 15f present the fine and multiscale (with no iterations) results of the test case. From the multiscale result it is clear that MS-DFM accurately captures the trend of the fine scale solution. Fig. 15d 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 ($n_{\text{finecells}}/n_{\text{coarsecells}}$). 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. Fig. 16 presents the coarse matrix dual-grid for these cases. Fig. 17 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 cells between the fractures are not influenced by any matrix dual-nodes (see Fig. 16b). 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 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. 18a–18c. The fine scale solution is presented in Fig. 18d 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.

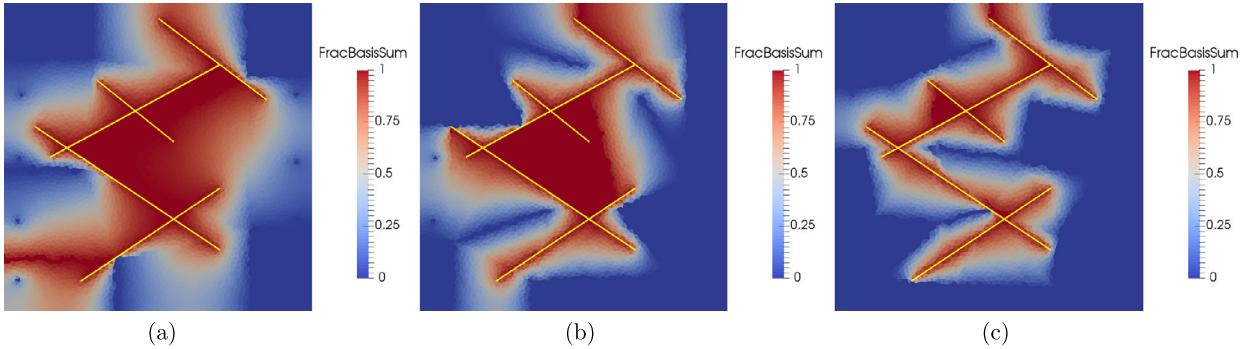


Fig. 17. Comparison of the sum of fracture basis functions in matrix domain, i.e., $\sum \Phi^{mf}$, for 3 levels of coarse grid refinement: (a) 5, (b) 12, and (c) 20 coarse matrix cells. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

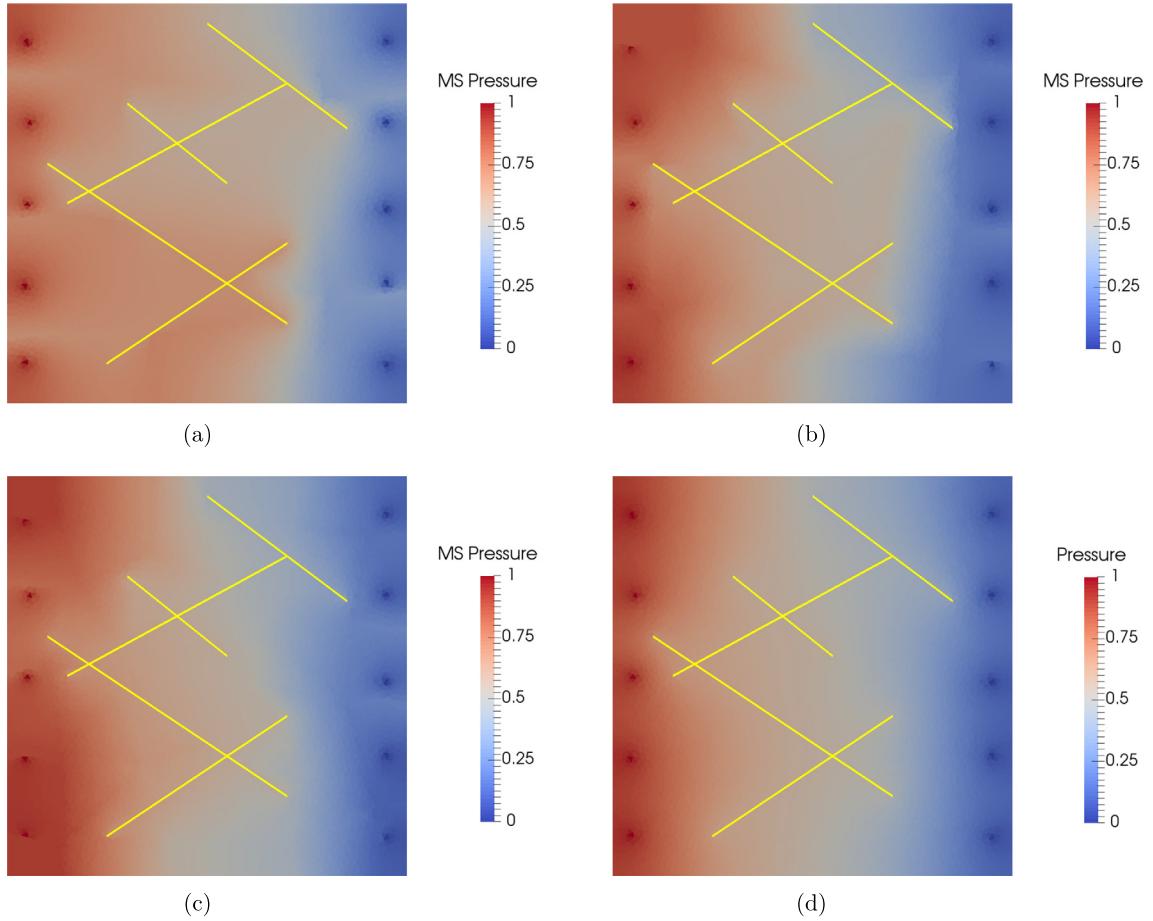


Fig. 18. Comparison of fine reference solution and MS-DFM solutions (with no iterations) of different levels of refinement: (a) 5, (b) 12 and (c) 20 coarse matrix cells with 2, 8 and 18 coarse fracture cells, respectively. Also shown is the fine-scale reference solution (d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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.

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,

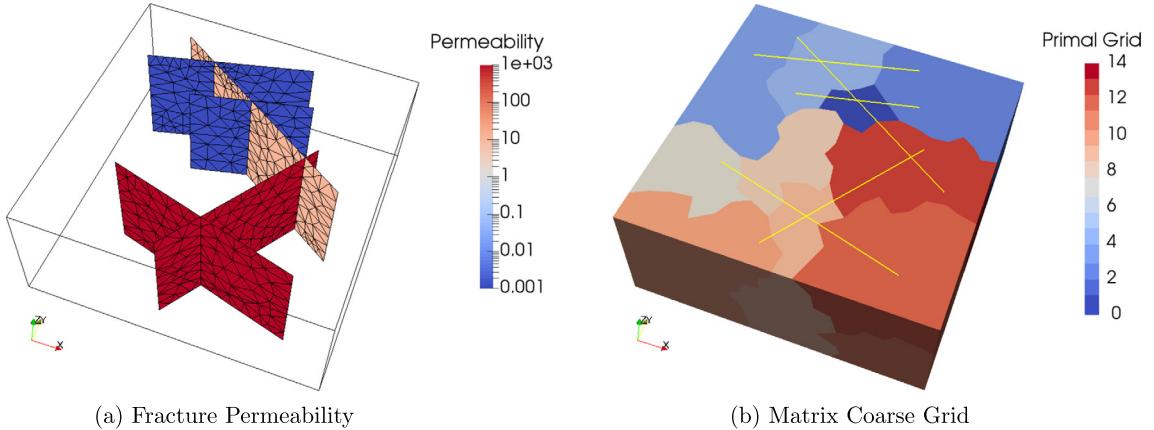


Fig. 19. 3D test case permeability map, and primal coarse grid for the matrix domain. Note that the matrix is homogeneous and is assigned non-dimensional permeability $k = 1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

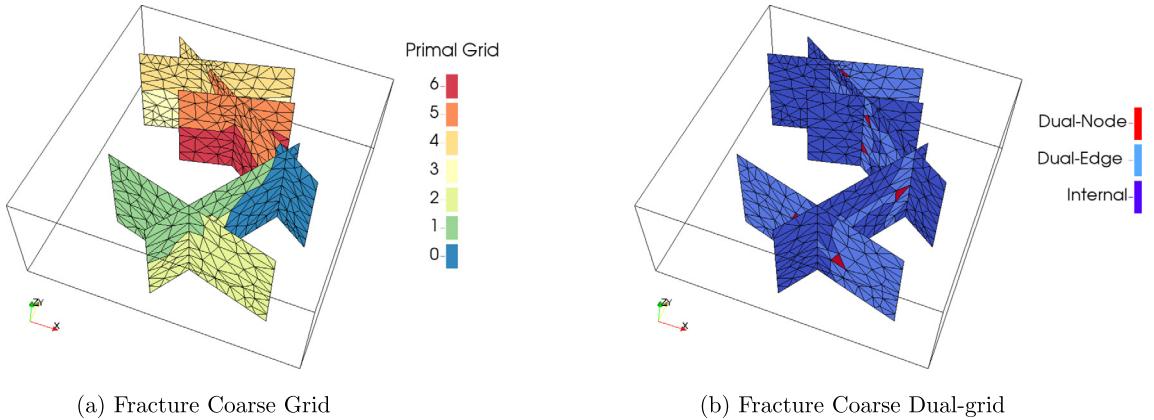


Fig. 20. Illustration of the fracture coarse grids for the 3D test case. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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. 19a. 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 Figs. 19b, 20a and 20b 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 $[x = 0.95, z = 0.05]$.

The MS-DFM result without iterations and converged reference solution are depicted in Fig. 21. Here, the MS-DFM approximation without iterations fails to correctly capture the effect of the sealing fractures. However, applying the iterative solving strategy, MS-DFM rapidly converges to the fine scale reference. By means of isobar plots, Figs. 22a to 22c 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. 22d.

6. Conclusions

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 iterations are employed to obtain the desired accuracy, which is typically much before full convergence is reached. In this case, locally mass conservative velocity field needs to be constructed before solving the transport equations. Future research will look into the application of MS-DFM on multiphase flow scenarios, along with error estimate and control studies. The proposed MS-DFM enables large flexibility in input fracture and matrix grids while at the same time only requiring geometrical data. As such, MS-DFM also provides the foundation for a unified DFM and EDFM framework where, depending on the relevant local physics, both methods can be adaptively used. Additionally, the method allows for a straightforward black box integra-

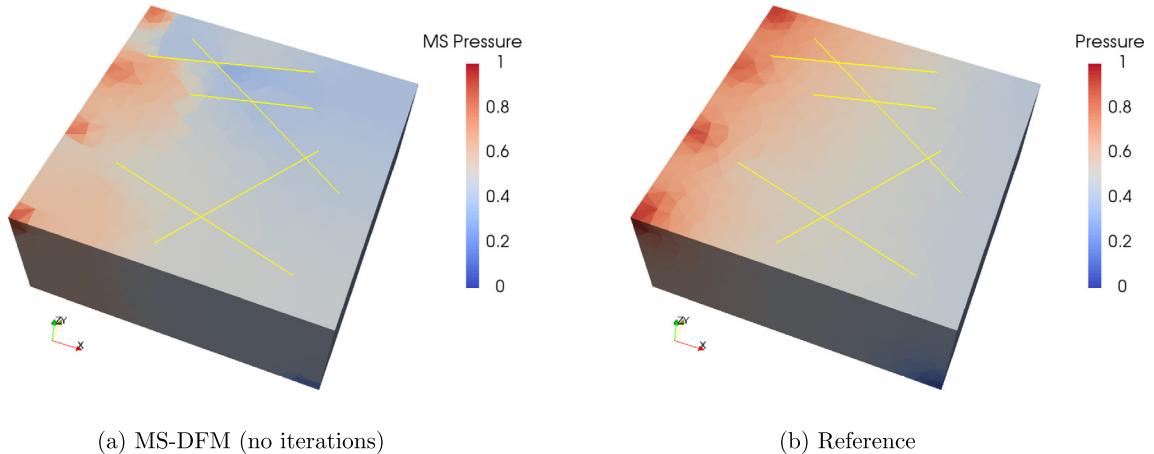


Fig. 21. Comparison between MS-DFM approximation with no iterations and the converged reference solution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

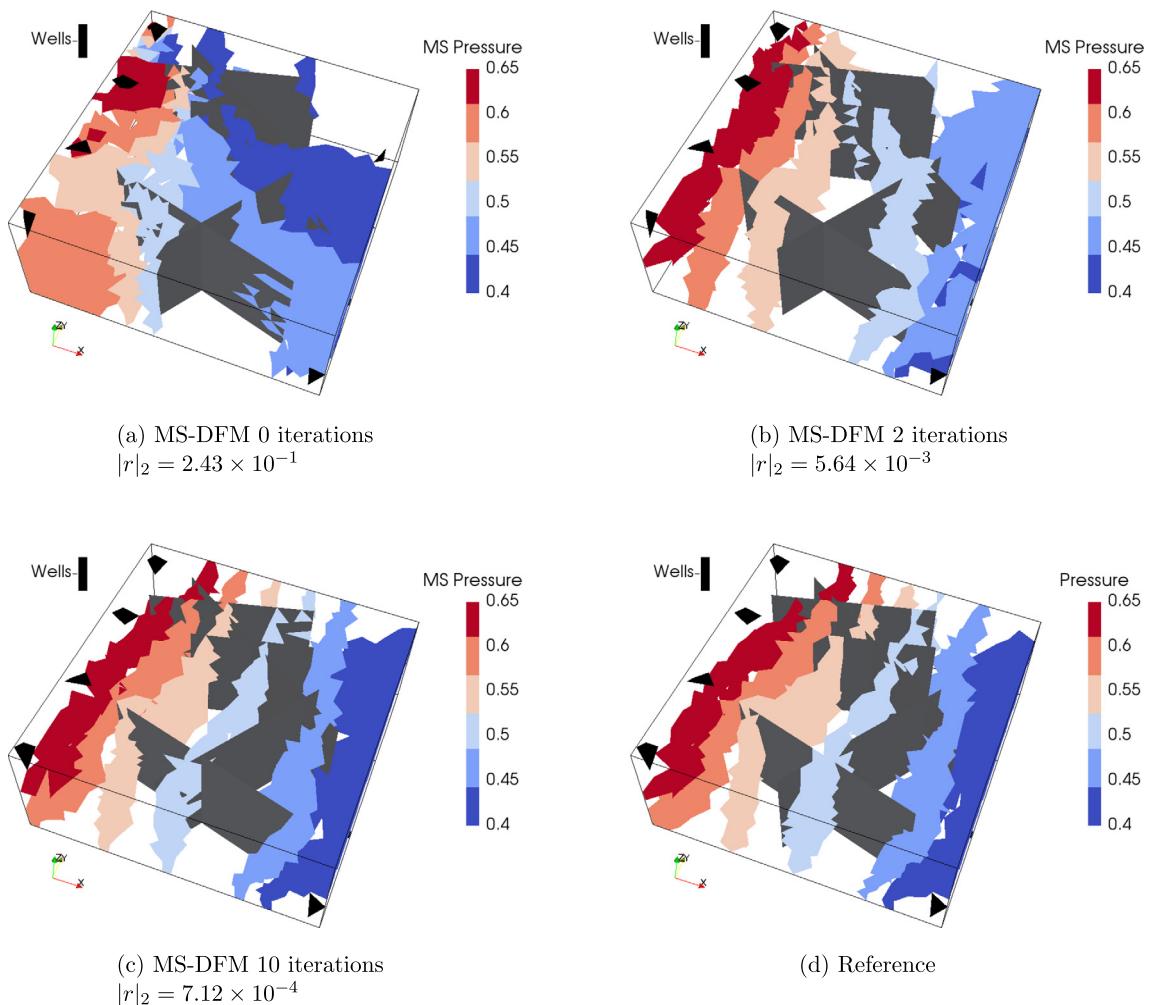


Fig. 22. Isobar plots illustrating convergence of the MS DFM solution of the 3D test case. Solutions after 0, 2 and 10 iterations are compared with the converged reference solution. Shown in gray surfaces are the fracture plates. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tion 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. These are subjects of future research. Furthermore, the method is proven to be accurate for 2D and 3D test cases. 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.

Acknowledgements

Financial support from Delft Advanced Reservoir Simulation (DARSim), the Stanford University Petroleum Research Institute for Reservoir Simulation (SUPRI-B), the Justus and Louise van Effen fund and Xodus Group are greatly acknowledged.

References

- [1] B. Berkowitz, Characterizing flow and transport in fractured geological media: a review, *Adv. Water Resour.* 25 (2002) 861–884.
- [2] L. Li, S.H. Lee, Efficient field-scale simulation of black oil in naturally fractured reservoir through discrete fracture networks and homogenized media, *SPE Reserv. Eval. Eng.* 11 (2008) 750–758.
- [3] H. Hajibeygi, D. Karvounis, P. Jenny, A hierarchical fracture model for the iterative multiscale finite volume method, *J. Comput. Phys.* 230 (2011) 8729–8743.
- [4] A. Moinfar, A. Varavei, K. Sepehrnoori, R.T. Johns, Development of an efficient embedded discrete fracture model for 3d compositional reservoir simulation in fractured reservoirs, *SPE J.* 19 (2014) 289–303.
- [5] Y. Efendiev, S.H. Lee, G. Li, J. Yao, N. Zhang, Hierarchical multiscale modeling for flows in fractured media using generalized multiscale finite element method, *Int. J. Geomath.* 6 (2015) 141–162.
- [6] M. Tene, M.A. Kobaisi, H. Hajibeygi, Multiscale projection-based embedded discrete fracture modeling approach (f-ams-pedfm), in: Proceedings of the 15th European Conference on the Mathematics of Oil Recovery, ECMOR XV, <http://dx.doi.org/10.3997/2214-4609.201601890>.
- [7] M. Tene, S. Bosma, M.A. Kobaisi, H. Hajibeygi, Projection-based embedded discrete fracture model (pedfm), *Adv. Water Resour.* 105 (2017) 205–216.
- [8] M. Karimi-Fard, L. Durlofsky, K. Aziz, An efficient discrete-fracture model applicable for general-purpose reservoir simulators, *SPE J.* 9 (2004) 227–236.
- [9] R. Ahmed, M.G. Edwards, S. Lamine, B.A.H. Huisman, M. Pal, Control-volume distributed multi-point flux approximation coupled with a lower-dimensional fracture model, *J. Comput. Phys.* 284 (2015) 462–489.
- [10] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig, A mixed-dimensional finite volume method for two-phase flow in fractured porous media, *Adv. Water Resour.* 29 (2006) 1020–1036.
- [11] S. Geiger-Boschung, S.K. Matthäi, J. Niessner, R. Helmig, Black-oil simulations for three-component, three-phase flow in fractured porous media, *SPE J.* 14 (2009) 338–354.
- [12] S. Pluimers, Hierarchical Fracture Modeling, Msc thesis, Delft University of Technology, The Netherlands, 2015.
- [13] B. Flemisch, I. Berre, W. Boon, A. Fumagalli, N. Schwenck, A. Scotti, I. Stefansson, A. Tatomir, Benchmarks for single-phase flow in fractured porous media, [arXiv:1701.01496](https://arxiv.org/abs/1701.01496), 2017.
- [14] T.Y. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *J. Comput. Phys.* 134 (1997) 169–189.
- [15] P. Jenny, S.H. Lee, H.A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, *J. Comput. Phys.* 187 (2003) 47–67.
- [16] Y. Efendiev, V. Ginting, T. Hou, R. Ewing, Convergence of a nonconforming multiscale finite element method, *SIAM J. Numer. Anal.* 37 (2000) 888–910.
- [17] Y. Efendiev, T.Y. Hou, Multiscale Finite Element Methods: Theory and Applications, Springer, 2009.
- [18] H. Hajibeygi, G. Bonfigli, M. Hesse, P. Jenny, Iterative multiscale finite-volume method, *J. Comput. Phys.* 227 (2008) 8604–8621.
- [19] H. Zhou, H.A. Tchelepi, Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, *SPE J.* 17 (2) (2012) 523–539.
- [20] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Algebraic multiscale linear solver for heterogeneous elliptic problems, *J. Comput. Phys.* 259 (2014) 284–303.
- [21] M. Tene, Y. Wang, H. Hajibeygi, Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media, *J. Comput. Phys.* 300 (2015) 679–694.
- [22] H. Zhou, S. Lee, H. Tchelepi, Multiscale finite-volume formulation for saturation equations, *SPE J.* 17 (2011) 198–211.
- [23] S.H. Lee, C. Wolfsteiner, 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 (2008) 351–366.
- [24] H. Hajibeygi, H.A. Tchelepi, Compositional multiscale finite-volume formulation, *SPE J.* 19 (2014) 316–326.
- [25] M. Cusini, C. van Kruijsdijk, H. Hajibeygi, Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media, *J. Comput. Phys.* 314 (2016) 60–79.
- [26] O. Møyner, K. Lie, A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids, *J. Comput. Phys.* 304 (2016) 46–71.
- [27] O. Møyner, K.-A. Lie, The multiscale finite volume method on unstructured grids, *SPE J.* 19 (2014) 816–831.
- [28] E. Parramore, G. Edwards, M. Pal, S. Lamine, Multiscale finite-volume cvd-mpfa formulations on structured and unstructured grids, *Multiscale Model. Simul.* 14 (2) (2016) 559–594.
- [29] T. Sandve, E. Keilegavlen, J. Nordbotten, Physics-based preconditioners for flow in fractured porous media, *Water Resour. Res.* 50 (2) (2014) 1357–1373.
- [30] J. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, S. Khataniar, Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs, *SPE J.* 16 (2011) 880–888.
- [31] M. Tene, M.S.A. Kobaisi, H. Hajibeygi, Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (f-ams), *J. Comput. Phys.* 321 (2016) 819–845.
- [32] S. Shah, O. Møyner, M. Tene, K. Lie, H. Hajibeygi, The multiscale restriction smoothed basis method for fractures porous media (f-msrsb), *J. Comput. Phys.* 318C (2016) 36–57.
- [33] J. Wallis, H.A. Tchelepi, Apparatus, method and system for improved reservoir simulation using an algebraic cascading class linear solver, US Patent 7,684,967, March 2010.
- [34] H. Zhou, H.A. Tchelepi, Operator based multiscale method for compressible flow, *SPE J.* 13 (2008) 267–273.
- [35] H. Si, Tetgen – A Quality Tetrahedral Mesh Generator and a 3d Delaunay Triangulator, Tech. Rep. version 1.4.3, Weierstrass Institute for Applied Analysis and Stochastics, 2011.
- [36] J. Shewchuk, Triangle – A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator, Tech. Rep. version 1.6, University of California at Berkeley, 2005.

- [37] K. Terekhov, Y. Vassilevski, Two-phase water flooding simulations on dynamic adaptive octree grids with two-point nonlinear fluxes, *Russ. J. Numer. Anal. Math. Model.* 28 (3) (2013) 267–288.
- [38] G. Karypis, Metis – Serial Graph Partitioning and Fill-Reducing Matrix Ordering, Tech. Rep. version 5.1.0, Karypis Lab, University of Minnesota, 2013.
- [39] C. Wolfsteiner, S. Lee, H.A. Tchelepi, Well modeling in the multiscale finite volume method for subsurface flow simulation, *Multiscale Model. Simul.* 5 (3) (2006) 900–917.
- [40] D. Peaceman, H. Rachford, The numerical solution of elliptic and parabolic differential equations, *J. Soc. Ind. Appl. Math.* 3 (1955) 28–41.