SPE 163649: The Multiscale Finite Volume Method on Unstructured Grids

Olav Møyner, Knut-Andreas Lie

Abstract

Finding a pressure solution for large-scale reservoirs that takes into account fine-scale heterogeneities can be very computationally intensive. One way of reducing the workload is to employ multiscale methods that capture local geological variations using a set of reusable basis functions. One of these methods, the multiscale finite-volume (MsFV) method is well studied for 2D Cartesian grids, but has not been implemented for stratigraphic and unstructured grids with faults in 3D. With reservoirs and other geological structures spanning several kilometers, running simulations on the meter scale can be prohibitively expensive in terms of time and hardware requirements. Multiscale methods are a possible solution to this problem, and extending the MsFV method to realistic grids is a step on the way towards fast and accurate solutions for large-scale reservoirs.

We present a MsFV solver along with a coarse partitioning algorithm that can handle stratigraphic grids with faults and wells. The solver is an alternative to traditional upscaling methods, but can also be used for accelerating fine-scale simulations. Approximate solutions computed by the new MsFV solver are compared with fine-scale solutions computed by a standard two-point scheme for grids with realistic permeability and geometries. The results show that the MsFV method is suitable for solving realistic permeabilities, but can fail for highly anisotropic systems. The implementation is a suitable framework for further experimentation with partitioning algorithms and MsFV variants. To achieve better precision, the implementation can use the MsFV method as a preconditioner for Arnoldi iterations using GMRES, or for smoothing cycles using Dirichlet Multiplicative Schwarz (DMS).

Introduction

Multiscale methods have been proposed as a way of bridging the gap in resolution between geological models (cell sizes: centimeters to decimeters in the vertical direction, meters to tens of meters in horizontal direction) and dynamic simulation models (cell sizes: meters to tens of meters in vertical direction, tens of meters to hundred of meters in horizontal direction). As an alternative to traditional upscaling techniques, multiscale methods (Efendiev and Hou 2009) can resolve fine-scale qualities with reduced computational complexity (Kippe et al. 2008) on highly detailed reservoir models by creating basis functions that relate the fine scale (geological model) to a coarser scale (dynamic model). Two methods have received particular interest in industry, the multiscale finite-volume (MsFV) methods (Jenny et al. 2003) and the multiscale mixed finite-element (MsMFE) method (Hou and Wu 1997). Both methods compute flow solutions using degrees-of-freedom associated with a coarse grid, but differ in the way they construct the multiscale basis functions that are used to account for fine-scale effects in the coarse-scale system and reconstruct approximate solutions on grids with fine or intermediate resolution. In the MsMFE method the basis functions are constructed by unit flow across faces in the coarse grid, whereas the basis functions in the MsFV method are computed using a dual-grid formulation with unitary pressure values at each vertex of the coarse blocks.

The MsFV method has been extended to a wide variety of problems in subsurface flow, including density-driven flow (Lunati and Jenny 2008) and compressible multiphase flow (Lunati and Jenny 2006; Zhou and Tchelepi 2008), by adding an extra set of correction functions. The method can also be employed in an iterative framework on its own (Hajibeygi and Jenny 2011) or as a preconditioner (Lunati et al. 2011), having a close relationship to domain-decomposition methods (Nordbotten and Bjørstad 2008). A particular advantage of the MsFV method, compared with multigrid and domain-decomposition methods, is that the multiscale method can reconstruct a conservative flux field at any iteration stage, which is crucial if the flux field is used to solve transport problems.

Application to unstructured grids is a remaining challenge for the MsFV method. Although the modeling approaches used by the industry today are predominantly structured, they typically lead to irregular and unstructured simulation grids. Very complex grids having unstructured connections and degenerate cell geometries arise naturally when representing structural framework like faults, joints, and deformation bands, and/or stratigraphic architectural characteristics like channels, lobes, clinoforms, and shale shale/mud drapes. Similarly, unstructured connections are induced when local grid refinement, structured or unstructured, is used to improve the modeling of (deviated) wells. With the exception of a highly idealized model of faults using 2D structured grids (Hajibeygi et al. 2011), the MsFV method has so far only been formulated and applied to regular Cartesian grids. Extending the MsFV formulation from uniform Cartesian to industry-standard grids with complex geometries/topologies and high aspect/anisotropy ratios is therefore a key step on the road to widespread adoption in industry. In particular, it is desirable to develop automated coarsening strategies that perform well for complex methods and preferably adapt to geological features and

well paths to ensure optimal accuracy for a chosen level of coarsening. This, in turn, requires robustness and a high degree of flexibility with respect to the size and shape of coarse blocks.

Multiscale finite element methods, and the MsMFE method in particular, have the required flexibility and can easily be defined for coarse grids in which each block consists of an (almost) arbitrary simply-connected set of polyhedral cells, see e.g., Aarnes et al. (2006, 2008); Natvig et al. (2011); Alpak et al. (2011); Lie et al. (2012). Whereas it has proved to be difficult to extend the original MsFV formulation (Jenny et al. 2003) to grids with unstructured connections, the more recent operator formulation (Zhou and Tchelepi 2008; Lunati et al. 2011) offers much of the flexibility needed. Using this formulation, the key challenges to extend the method to grids with unstructured connections and irregular geometries are to develop: (i) automated and robust methods for constructing primal and dual coarse grids, and (ii) appropriate localization conditions for the basis functions that are robust to large aspect ratios and strong variations in media properties at the faces of the coarse blocks. The second problem has recently been addressed by Wang et al. (2012). Herein, we mainly attack the first problem and present a formulation of the MsFV method for general unstructured grids and a corresponding set of coarsening methods that can create the required primal/dual grids for a reasonable class of stratigraphic grids. Our approach for coarsening is to first create a primal coarse grid using techniques developed for the MsMFE method, and then create the accompanying dual based on geometrical considerations. The alternative approach of first creating the dual will be pursued elsewhere.

The software used to create all examples presented in the following, is implemented as a separate module in the open-source Matlab Reservoir Simulation Toolbox (MRST 2012) and can be freely downloaded and used within the GPL 3.0 software license.

The Multiscale Mixed Finite Volume Method

To keep the presentation of new ideas as simple as possible, we will in the following only consider incompressible flow and neglect the effects of capillarity and gravitational forces; these effects can easily be included as discussed in the literature. Combining mass conservation, $\nabla \cdot \vec{v} = q$, with Darcy's law, $\vec{v} = -\mathbf{K}\lambda_t \nabla p$, we obtain a Poisson-type, elliptic pressure equation

$$-\nabla \cdot (\mathbf{K}\lambda_t \cdot \nabla p) = q \tag{1}$$

for the fluid pressure p and the total velocity v. Here, q denotes fluid sources, $\mathbf{K}$ is the permeability tensor, and $\lambda_t = \sum_\alpha \lambda_\alpha$ is the total mobility, where the mobility λ_α of phase α equals the ratio of the relative permeability k_α and the viscosity μ_α of the phase.

The MsFV approach can be informally described as first computing flow problems localized to each coarse block to define a set of basis functions for the pressure on some set of coarse blocks which are then used as a set of pressure basis functions. These basis functions can then be used to construct a coarse pressure system, whose solution can be extended to the fine grid using the subresolution of the pressure basis functions. Once a solution has been obtained, the error can be reduced by iteratively constructing a set of *correction functions* (Hajibeygi and Jenny 2011; Lunati and Jenny 2008) that take into account fine-scale effects of wells, compressibility, gravity, etc. At any time, another set of local problems with Neumann boundary conditions can be solved in each coarse block that result in conservative fluxes on the fine scale, if needed. For this paper, we will focus on the initial pressure solution since this is where unstructured grids add additional complexity. The other parts of the method that are not covered herein can be used as before.

Fine-scale discretization. A multiscale method will not have better accuracy or contain less numerical artifacts than the method used to discretize the flow equations on the fine scale. In industry, the predominant discretization approach is to use a two-point flux approximation (TPFA) method (i.e., the finite-volume counterpart of a seven-point finite-difference stencil on Cartesian grids in 3D), even though it is well known that this method is only consistent for K-orthogonal grids and will therefore not converge in the general case. On the other hand, this method is guaranteed to be monotone, unlike consistent (linear) discretization methods like the multipoint flux approximation (MPFA) schemes, mimetic finite differences, and mixed finite elements.

To develop the multiscale formulation, we assume that the fine-scale discretization of Eq. 1 results in a linear system on the form

$$A_h \mathbf{p}_h = \mathbf{q}_h, \tag{2}$$

where A_h corresponds to a set of mass balance equations for the cell-center pressures $\mathbf{p}_h$ with the source terms being collected in $\mathbf{q}_h$. Although the general form Eq. 2 includes consistent discretizations like MPFA schemes, we will in the following only use the standard TPFA method in all numerical examples.

Grids and orderings. As in all conventional pressure solvers, a fine-scale discretization is imposed on a grid Ω_h consisting of a collection of cells $\{c_i\}_{i\in\mathcal{I}}$ that covers the whole computational domain Ω. In addition to the fine grid Ω_h, the MsFV method requires two different coarse grids to be defined. The *primal coarse grid*, denoted as Ω_H, consists of coarse blocks B_j that are composed of fine cells $c_i \in \Omega_h$ such that $\Omega = \cup_{j\in[1,N]} B_j$, with every fine cell belonging to one and only one block B_j. The coarse grid need not be defined explicitly to implement the MsFV method, but to get conservative flux a categorization of each c_i to some B_j is required.

The *dual coarse grid* $\tilde{\Omega}_H$ has corners corresponding to the block centers of Ω_H. Because Ω_H is defined by fine cells, its edges run alongside the fine-cell edges. The dual grid $\tilde{\Omega}_H$, however, has edges running through fine cells so that we can define a set

of fine cells as edge cells. This concept is central to the operator formulation seen in (Lunati et al. 2011; Lunati and Lee 2009), wherein all fine cells c_i are categorized according to their positions in $\tilde{\Omega}_H$. The categories can be explained as follows:

- **Nodes** ($\mathcal{I}_n$): Cells containing the centers of Ω_H. For 3D Cartesian grids, these are points where three coarse faces of $\tilde{B}_j$ intersect.
- **Edges** ($\mathcal{I}_e$): Cells on the edge of the faces of $\tilde{\Omega}_H$. For 3D Cartesian grids, these are the cells common to two faces of at least one block $\tilde{B}_j$.
- **Faces** ($\mathcal{I}_f$): Cells on a faces of the dual coarse grid $\tilde{\Omega}_H$, but not on the edges of the face. For 3D Cartesian grids, this would be equivalent with being on a single face for some $\tilde{B}_j$. For 2D domains this category is omitted.
- **Inner** ($\mathcal{I}_i$): Cells not on the faces of $\tilde{\Omega}_H$.

Previous formulations of the MsFV method have only used three categories (i.e., node, edge+face, and inner cells), from which we get the following decomposition

$$\mathcal{I} = \mathcal{I}_n \cup \mathcal{I}_{e+f} \cup \mathcal{I}_i. \tag{3}$$

We refer to the method using decomposition Eq. 3 as MsFV (3-step) to distinguish it from the method stemming from the alternate decomposition into four categories (node, edge, face, and inner),

$$\mathcal{I} = \mathcal{I}_n \cup \mathcal{I}_e \cup \mathcal{I}_f \cup \mathcal{I}_i, \tag{4}$$

which we refer to as MsFV (4-step). The two decompositions are equivalent for two dimensional problems and we will therefore only present a detailed derivation of the 4-step method since we will focus on 3D problems. For more details on the general stages involved both for the 3-step and 4-step variants, see (Lunati et al. 2011; Lunati and Lee 2009; Møyner 2012).

Permuting the system matrix. Once all cells have been categorized, we construct a $N \times N$ permutation matrix P that reorders Eq. 2 so that the unknowns are grouped by categorization of their corresponding cells:

$$P\mathbf{p}_h = \mathbf{p} = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{p}_f \\ \mathbf{p}_e \\ \mathbf{p}_n \end{bmatrix}, \qquad PA_hP^T = A = \begin{bmatrix} A_{ii} & A_{fi} & 0 & 0 \\ A_{if} & A_{ff} & A_{fe} & 0 \\ 0 & A_{ef} & A_{ee} & A_{ne} \\ 0 & 0 & A_{en} & A_{nn} \end{bmatrix}. \tag{5}$$

Note the indices in Eq. 5: block A_{kl} represents the influence from cells of category *k* to the cells of category *l* in the system of mass balances. This system is reduced to upper block-triangular form by removing connections so that edges only depend on the values of nodes, the faces only on the edges, and so on. To remove the connections from category *k* to *l*, block A_{kl} is removed and A_{ll} is modified by removing the mass loss corresponding to the disconnected cells,

$$(M_{ll})_{rr} = (A_{ll})_{rr} + \sum_s (A_{kl})_{rs}. \tag{6}$$

Once all lower-diagonal blocks have been eliminated in this way, we obtain the multiscale system matrix,

$$\begin{bmatrix} A_{ii} & A_{if} & 0 & 0 \\ 0 & M_{ff} & A_{fe} & 0 \\ 0 & 0 & M_{ee} & A_{ne} \\ 0 & 0 & 0 & M_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ \mathbf{p}_f \\ \mathbf{p}_e \\ \mathbf{p}_n \end{bmatrix} = \begin{bmatrix} \mathbf{q}_i \\ \mathbf{q}_f \\ \mathbf{q}_e \\ \mathbf{q}_n \end{bmatrix}, \tag{7}$$

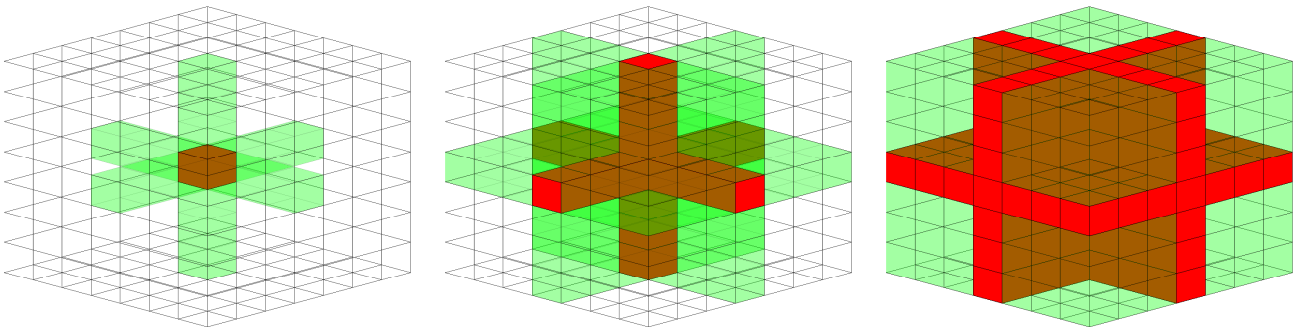
which can be solved using backward substitution. If we for a while assume that the values $\mathbf{p}_n$ in the cells of category 'node' are known, it follows immediately from Eq. 7 that the unknown pressure values in the 'edge' cells are given by $M_{ee}\mathbf{p}_e + A_{en}\mathbf{p}_n = \mathbf{q}_e$, and hence can be computed by first extrapolating values from 'node' cells $A_{en}\mathbf{p}_n$, and then solving for the rest of the category, $M_{ee}^{-1}(\mathbf{q}_e - A_{en}\mathbf{p}_n)$. Similarly, the face and inner unknowns can be computed by first extrapolating values from the previous category and then inverting the diagonal matrix block. This procedure is illustrated in Figure 1. Altogether, the solution has the form

$$\mathbf{p} = B\mathbf{p}_n + C\mathbf{q}, \tag{8}$$

where the matrices B and C are defined as follows:

$$B = \begin{bmatrix} A_{ii}^{-1}A_{if}M_{ff}^{-1}A_{fe}M_{ee}^{-1}A_{en} \\ M_{ff}^{-1}A_{fe}M_{ee}^{-1}A_{en} \\ M_{ee}^{-1}A_{en} \\ I \end{bmatrix}, \qquad C = \begin{bmatrix} A_{ii}^{-1} & -A_{ii}^{-1}A_{if}M_{ff}^{-1} & A_{ii}^{-1}A_{if}M_{ff}^{-1}A_{fe}M_{ee}^{-1} & 0 \\ 0 & M_{ff}^{-1} & -M_{ff}^{-1}A_{fe}M_{ee}^{-1} & 0 \\ 0 & 0 & M_{ee}^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9}$$

Here, we see that the matrix B consists of the basis functions that extend the coarse pressure solution $\mathbf{p}_n$ to the fine grid, whereas C consists of the correction functions that handle fine-scale behavior of fluid sources (wells).



(a) Node to edge (b) Edge to face (c) Face to inner

Fig. 1— Illustration of the three extrapolation stages of the 4-step MsFV formulation. In each plot, the previous step is shown in red, with the current in green.

Coarse-scale system. When backwards substituting the system, we need to find values for the coarse node pressure $\mathbf{p}_n$. Unfortunately, we cannot simply find M_{nn} in the same manner as for example M_{ee}: If we take the node influence on the other nodes directly, we will get an empty matrix since the coarse nodes are not neighbors on the fine grid. To find a volume-conservative coarse-scale solution accounting for-fine scale heterogeneities, some further work is needed.

The system for the 'node' cells should correspond to a coarse-scale system defined over the primal coarse grid. Such a system can be defined if we in each coarse block sum up the mass balance equations for all the cells that belong to the block. To this end, we introduce a summation matrix χ that has one row χ_j for each primal coarse block B_j, defined such that element $\chi_{ij} = 1$ if cell c_i belongs to B_j and $\chi_{ij} = 0$ otherwise. Applying this summation operator to the reordered system and inserting Eq. 8, we obtain

$$\chi A\mathbf{p} = \chi A\big(B\mathbf{p}_n + C\mathbf{q}\big) = \chi\mathbf{q} \quad \Longrightarrow \quad M_{nn}\mathbf{p}_n = \chi AB\mathbf{p}_n = \chi\big(I - AC\big)\mathbf{q} = \mathbf{q}_n. \tag{10}$$

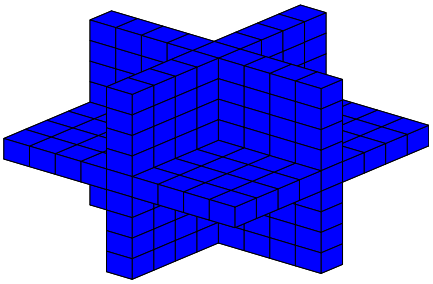
This allows us to construct a coarse pressure system which can be combined with the basis functions from Eq. 8 to construct a fine-scale pressure solution. For the nature of the operators used in these systems, we again refer to (Lunati et al. 2011; Lunati and Lee 2009). While additional steps are required to create iterative variants and construct conservative flux, these steps are the same for all variants of the method and have been covered thoroughly in earlier articles om the MsFV method.

Necessity of 4-step decomposition i 3D The algorithmic difference between the 3-step and 4-step decompositions lies in how pressure values are extrapolated from 'node' cells to cells on the faces of the dual coarse blocks. In the 3-step method, this extrapolation is done in one step, whereas the 4-step method uses two steps, first from nodes to edges and then from edges to faces. The two methods will therefore, in general, give different results when applied to the same problem.

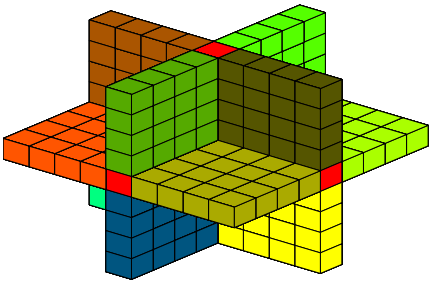
The need for an additional decoupling step when making the move to 3D should be obvious if we consider a regular 3D Cartesian grid. Figure 2 shows the faces of the dual coarse grid limited to a single primal coarse block. Altogether, there are 217 cells that lie on the faces of the dual grid. For the 4-step method, the center cell will be a 'node', the 24 red cells are 'edge' cells, while the remaining 192 cells are 'face' cells. In this simple example, it is easy to see that the 'face' cells can be organized as 12 patches of 4×4 cells, where the cells inside each patch are interconnected but not connected to cells in the other patches. This means that the diagonal block M^4_{ff} from the multiscale system matrix in Eq. 7 can be reordered into a block-diagonal form as shown in the lower-right plot of Figure 2. In the general case, one can view the M_{ff} matrix as an undirected graph, use a standard topological sort to identify cycles (i.e., group of cells that are connected to each other through some path in the system matrix), and then reorder M^4_{ff} into block-diagonal form based on the cycles. Each cycle corresponds to a diagonal block and contains cells that must be inverted simultaneously in a direct method. Since block-diagonal systems can be solved as a series of independent systems in a direct method, this makes this decomposition fully parallel. For the 3-step method in 3D, all cells, except for the center 'node' cell, are categorized as 'e+f' in Eq. 3 and are connected in a single cycle. The M^3_{ff} matrix has a more complex sparsity pattern, a higher condition number, and cannot be reordered, and will therefore be significantly more expensive to invert. This is the main motivation behind using a 4-step decomposition in 3D.

Generation of coarse grids

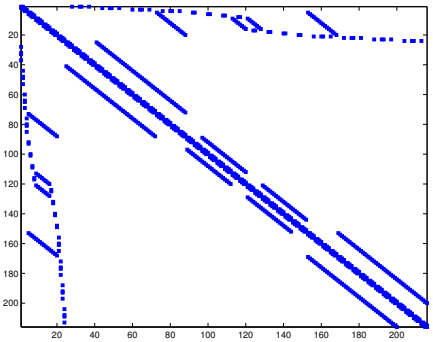
As explained above, the MsFV method requires two different coarse grids: a primal grid on which one formulates the global coarse-scale equations and a dual grid on which one computes the basis functions that are later used to determine the coarse-scale pressure unknowns associated with the primal coarse grid, see Figure 3. While having two sets of coarse grids is useful for getting both conservative flow and continuous pressure in the MsFV, such grids can be difficult to implement for general, unstructured grids often encountered in real world scenarios. In particular, whereas primal coarse grids are used in upscaling and are readily available in most simulators, dual coarse grids are less common and significantly more difficult to generate.



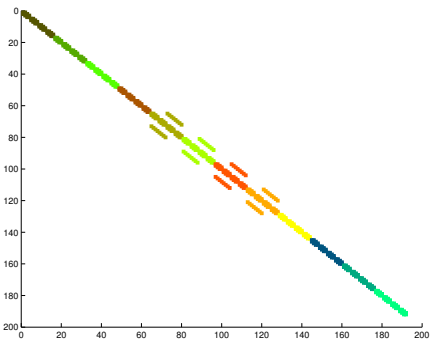
(a) 'Face+edge' cells for the 3-step method



(b) 'Face' and 'edge' cells for the 4-step method



(c) Sparsity pattern for the face system M_{ff}^3



(d) Sparsity pattern for the face system M_{ff}^4

Fig. 2— Comparison of the extrapolation stages for the 3-step and 4-step MsFV methods. The upper plot shows the faces of the dual grid limited to a single coarse primal block. The lower plot shows the sparsity pattern of the corresponding face systems reordered to block-diagonal form and colored by cycle index. For the 4-step method, the edge system corresponding to the red cells are not included in the sparsity patterns.

Previous publications on the MsFV method have only considered regular Cartesian grids, for which a regular primary coarse grid, and its shifted dual, can be trivially obtained by introducing the tensor product of a load-balanced linear distribution along each axial direction. The same approach can be applied to partition the indices of stratigraphic grids that have a uniform Cartesian topology. However, most real-life models will have unstructured connections, degenerate cell geometries, and inactive cells, which all make the process of defining a reasonable pair of dual-primal coarse grids difficult. To this end, a key difficulty is to ensure that 'node' cells are connected through 'edge' cells, and that 'edge' cells are connected through 'face' cells. If not, it will be very difficult to compute basis functions and extrapolate pressure values between the different cell categories.

A lot more effort has gone into formulating the MsMFE method on unstructured grids and developing good coarsening methods (Aarnes et al. 2006; Kippe et al. 2008; Natvig et al. 2011; Hauge 2010; Alpak et al. 2011; Natvig et al. 2012). Extensive numerical experiments show that whereas introducing a 'black-box' regular partition may be sufficient to get acceptable accuracy in many cases, improved robustness and increased accuracy can be observed if the coarse grid is adapted to well paths and reservoir heterogeneity. However, formulating grids for the MsMFE method that both give high accuracy and reduced runtime requires some user experience. We expect that the same will be true for the MsFV method. To make the method more robust and accessible to non-expert users, an important task for future research is therefore to develop automated or semi-automated methods for performing the internal grid coarsening and a set of guidelines for setting the necessary parameters, see e.g. (Aarnes et al. 2008; Hauge 2010).

In the rest of this section, we will present methods that can be used to generate basic dual/primal coarse grids for a reasonably large class of stratigraphic grids.

Primal grid. A prototype of the MsFV method for stratigraphic and unstructured grids was implemented in the open-source Matlab Reservoir Toolbox (MRST 2012). All grids in MRST are assumed to consist of a set of nonoverlapping polyhedral cells, where each cell can have a varying number of planar faces that match the faces of the cell's neighbours. Grids with non-matching faces, arising for instance at a fault in the stratigraphic formate, are therefore converted into matching grids by splitting non-matching faces into a set of matching (sub)faces. All grids are stored in a general format in which we explicitly represent cells, faces, and vertices and the connections between cells and faces (Lie et al. 2012).

Beyond an implementation of the MsMFE method, the toolbox has modules for upscaling and for creating non-uniform coarse grids that can be constrained to geological features like high/low permeability structures, facies, stone types, or saturation regions, or adapted to flow indicators of various type. Generating primal coarse grids can therefore be done in several different and very flexible ways. Routines for generating coarse grids depend only on a partition vector containing the categories of all fine cells.

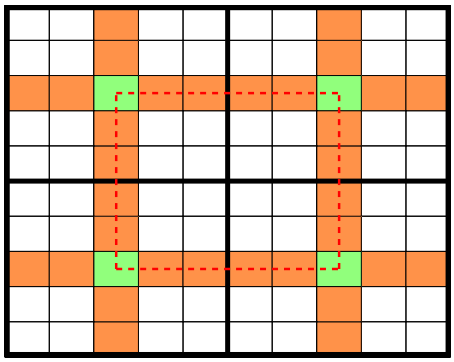


Fig. 3— A coarse grid, outlined in thick black lines along with the corresponding dual grid shown in orange, with a single dual block emphasized in red.

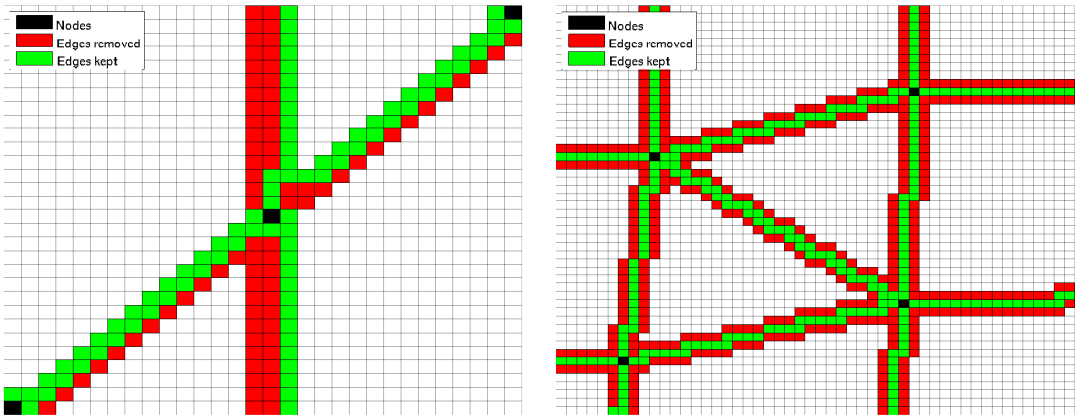
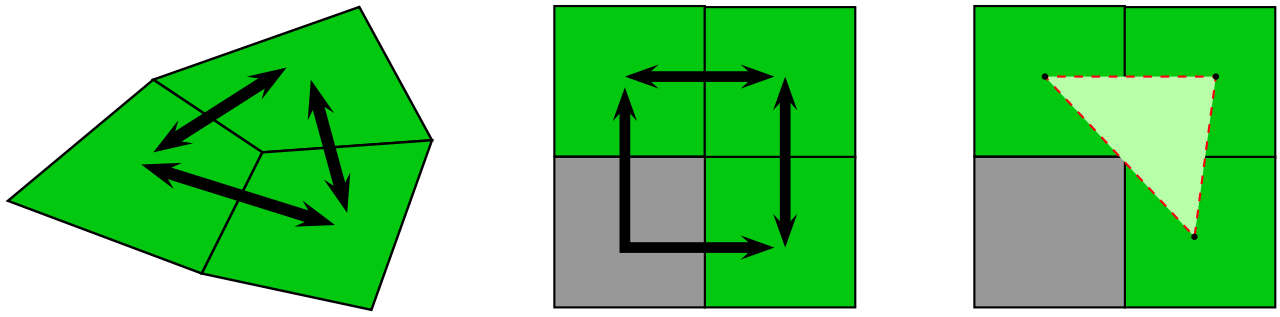


Fig. 4— The thickness of the dual grid can be adjusted by a culling algorithm, here shown on a small subset and a bigger subset in 2D of grid intentionally made extra thick.

Dual grid. There are three main approaches to generating a dual grid algorithmically: The first approach is, as explained above, to exploit that many grids have a structured topology. Using a logical partitioning scheme in index space has the advantage of being easy to implement and requires no lookup of actual node positions. For Cartesian grids, using logical partitioning to create a dual grid is trivially implemented by shifting the primal grid in each of the axial directions. The second approach is to use geometrical data when partitioning, using actual node coordinates and distances between them to generate a grid. This will require lookups in data structures, which can be quite costly depending on the data structure of the reservoir simulator used. The third approach is to use topological data by traversing the system graph and using this to construct suitable grids. This has the advantage of providing the algorithm with the actual connections used when solving the linear systems.

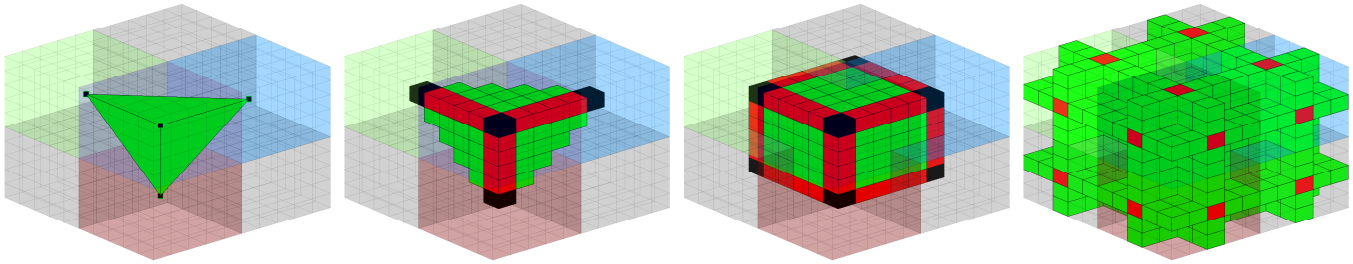
To produce working dual grids for unstructured grids, we first employ a geometrical algorithm to produce a dual grid which overestimates the number of edge cells, but ensures connections over the boundaries of coarse blocks. This dual grid is then post-processed by an algorithm that uses the system topology to strip away superfluous edge cells. To motivate our algorithm, we first consider a simple 2D case. Here, the dual grid is defined as the grid created by connecting the cell centers of the primal grid. A simple 2D grid consisting of four coarse blocks is shown in Figure 3. Generating a 2D dual grid is quite simple: Simply take pairs of coarse neighbor blocks and draw a line between their centroids, i.e., the dashed red line connecting two green cells in the figure. If a cell is intersected by one of these lines, it will be categorized as an edge cell (red cells in the figure).

The same approach can be applied to more general geometries/topologies of the fine and coarse grid. First, we use some threshold to associate all cells that lie within a certain distance to a line between two 'center' cells to the corresponding edge. Once a dual grid has been generated, it may or may not contain an excess of cells in the edge categories. This makes the solution less prone to errors, as the reduced system is less sensitive to single cells with high variations in permeability, but the linear systems corresponding to the edges will be larger. A culling function can be used to change remove excess cells from the edges: While some convergence criterion, change the category of 'edge' cells to 'inner' without reducing the number of cycles in A_{ii} and avoid connecting inner nodes to other categories than 'face'. By using the system graph based on the discretizations to define connections, this can be done quickly while ensuring that the final linear equation sets remain solvable. For an example of this, see Figure 4.



(a) (b) (c)

Fig. 5— Two neighbor sets – (a) shows an unstructured grid in which all blocks are neighbors of each other, while (b) shows a Cartesian subset in which two of the neighbors are connected via a fourth coarse block shown in gray. (c) shows how a triangle (red) is constructed from the centers of the neighbors.



(a) (b) (c) (d)

Fig. 6— From a block and three neighbors, three planes are generated (a). Once the planes have categorized the cells, the resulting nodes (black), edges (red) and faces (green) are determined (b, c). Adding virtual coarse blocks to the outside of the domain (d) lets the dual grid reach the boundary faces.

Defining a dual grid for 3D grids with complex geometries is nontrivial; Møyner (2012) presents several approaches that work in special cases. Herein, we present a more general approach that starts by defining sets of *common neighbors* and letting the partition algorithm work on these sets to produce the final grid. A set of *common neighbors* will be defined as three coarse blocks chosen so that every coarse-block is neighbor to at least one of the others and neighbors the other via some block not in the set. This allows us to cover both Cartesian-like neighborhood structures as well as unstructured grids, as can be seen in Figure 5. A triangle is then constructed between the centers of these neighbors as shown in the rightmost part of Figure 5. The left plot in Figure 6 shows the three planes defined for the three triplets of common neighbors that can be defined from the upper corner cell.

Once triangles have been generated for all triplets of common neighbors, any cells intersected by each plane will be added to the 'face' category $\mathcal{I}_f$. Any cells that are on the edge of three or more triangles are on the path between two direct neighbors and will be categorized as on the edges ($\mathcal{I}_l$). For our 3D example, the resulting categories from the planes previously discussed can be seen in the middle parts of Figure 6.

While not always stated when discussing the MsFV method, having dual edges that touch the boundary is important to be able to impose boundary conditions other than no-flow. The observant reader may have noticed that our algorithm only handles the domain between coarse centroids and as such will not create a dual grid capable of handling boundary conditions. Two alternatives exist to solve this problem: Either add virtual coarse blocks outside of the domain which do not contain any fine cells, or simply use the coarse-face centroids on the boundary to construct additional triangles. The rightmost plot of Figure 6 shows how this enables the rest of the dual grid to be constructed, and in Figure 7 the edge cells for a subset of a PEBI grid is shown.

Numerical Experiments

In this section, we apply the MsFV method to a set of models with increasing degree of topological and geometrical complexity. The first example is a corner-point model with a single fault and highly heterogeneous petrophysical properties. The second example is a synthetic model with a curved fault represented using 2.5D prismatic or PEBI grids. The third example uses the grid of a real-field model from the Norwegian Sea. In all three examples, we keep the flow physics as simple as possible since the main goal of this paper is to demonstrate that the capability of the MsFV method to handle realistic grids.

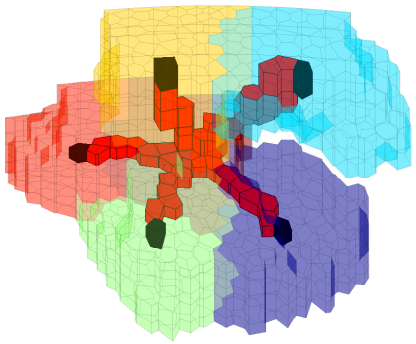
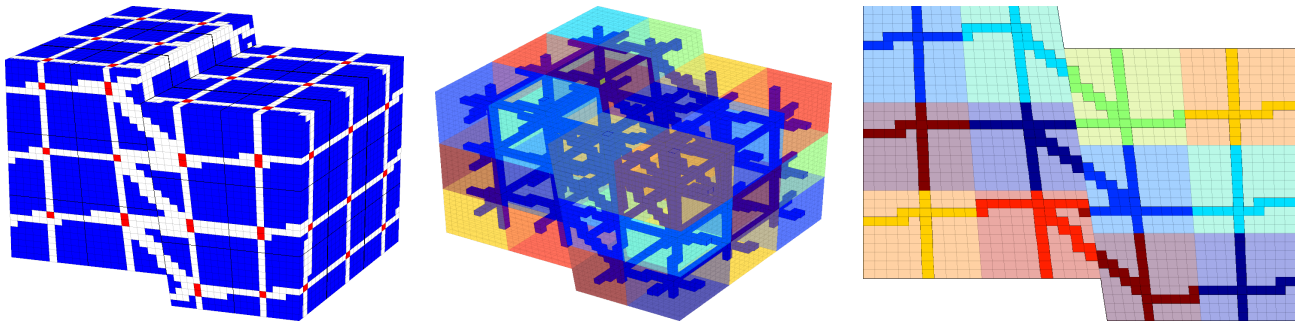


Fig. 7— By partitioning a subset of an unstructured PEBI grid, we can see the edge cells extruding from a single coarse block. Note that if the coarse block has more neighbors than the node cell has faces, we risk reduction in cycle count for the edges because the connections will overlap.



(a) (b) (c)

Fig. 8— Primal and dual coarse grids generated for a $40 \times 30 \times 31$ realization of the single-fault model. The plot in (a) shows the 'inner' cells (blue), the 'face' cells (white), the 'edge' cells (red), and the primal coarse grid (thick line) for a subset of the whole model. Plot (b) shows the different blocks in the primal $4 \times 3 \times 3$ coarse grid colored in different transparent colors to show the 'edge' cells inside, colored in blue. Plot (c) shows a side-view of the same plot, in which the 'edge' cells have been given a different color for each primal block.

To measure the discrepancy between fine-scale and multiscale solutions we will use scaled L_∞ and L_2 norms,

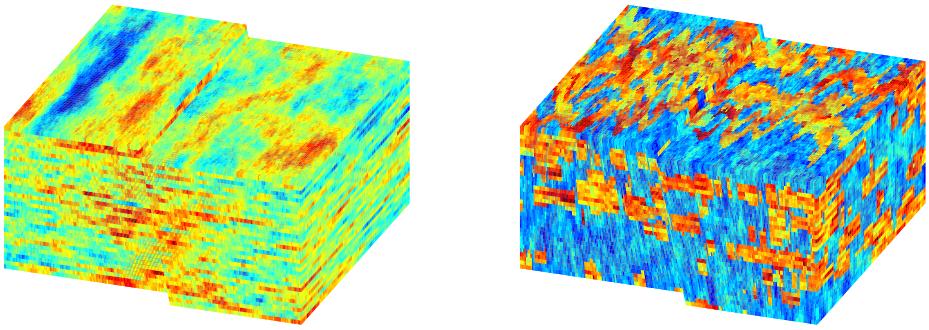
$$\|\mathbf{u}\|_\infty = \frac{\|\mathbf{u}_{fs} - \mathbf{u}_{ms}\|_\infty}{\|\mathbf{u}_{fs}\|_\infty}, \qquad \|\mathbf{u}\|_2 = \frac{\|\mathbf{u}_{fs} - \mathbf{u}_{ms}\|_2}{\|\mathbf{u}_{fs}\|_2} \tag{11}$$

where $\mathbf{u}$ is either the vector of cell pressures or fine-scale fluxes.

When the test problems include wells these are handled by the inclusion of correction functions.

A single fault. We consider a simple rectangular domain that contains a sloping fault across which there has been a significant displacement. The structure is modeled using a corner-point grid in which the pillars are vertical at the east and west sides of the domain and gradually become more skew to align with the fault in the middle of the domain. Figure 8 shows a realization of the structural model that has been partitioned into a primal and a dual coarse grid. To partition across the fault, coarse blocks divided by the fault are split in two to ensure that the primal blocks are relatively homogenous in petrophysical properties, which is one of the assumptions behind a coarse pressure system in the first place. At the fault, almost all the coarse blocks are connected to two blocks on the opposite side of the fault because of the displacement. As a result, four of the primal blocks shown in the side-view plot will have five edges emanating from the center ('node') cell instead of the usual four edges seen for blocks having only logically Cartesian neighbors.

Flow is driven from left to right by imposing Dirichlet boundary conditions of 500 bar and 250 bar along the left and right faces of the model. To investigate the accuracy of the MsFV method, we consider three different sets of petrophysical data: isotropic, homogeneous permeability equal 100 mD and two different subsamples of from Model 2 from the 10th SPE Comparative Solution Project (Christie and Blunt 2001), which is a geo-cellular model with highly heterogeneous petrophysical properties that was originally created as a challenging benchmark case for upscaling methods. Later, subsets of this model have become popular tests for multiscale and other numerical methods. In particular, the various formulation of the MsFV method have been tested extensively on layers of the full model. Herein, we use permeability sampled from the upper 35 layers of the model, which



Tarbert layers on faulted grid Upper Ness layers on faulted grid

Fig. 9— The same faulted grid with permeability from the SPE10 dataset.

Table 1— Discrepancies between the fine-scale and the MsFV solution computed for two different permeability distributions and two different structural models (Cartesian box and single-fault model from Figure 8) realized on a $60 \times 220 \times 35$ fine grid that is partitioned into a $44 \times 12 \times 7$ coarse grid.

Model		Error in pressure		Error in flux	
Permeability	Fault	L_2	L_∞	L_2	L_∞
Homogeneous	No	1.74858e-12	2.00261e-12	1.55120e-11	1.34408e-10
Homogeneous	Yes	1.35204e-03	3.01027e-03	4.42839e-02	3.89119e-01
Tarbert	No	2.43395e-02	1.84410e+00	4.91018e-01	2.27574e+00
Tarbert	Yes	1.68418e-01	2.26869e+01	1.68660e+00	3.93173e+00
Upper Ness	No	1.59527e+00	1.02909e+02	2.81082e+01	1.56371e+02
Upper Ness	Yes	6.14998e+01	3.35768e+03	3.36793e+02	2.04909e+03

correspond to a shallow-marine Tarbert formation with smoothly varying permeability that follows a lognormal distribution, and permeability sampled from the bottom 35 layers of the model, which correspond to a fluvial Upper Ness formation that consists of an intertwined set of high-permeable channels on a low-permeable background, see Figure 9.

Table 1 reports the discrepancies between the fine-scale and MsFV pressure and flux solutions measured in the relative L_2 and L_∞ norms. For comparison, we also include simulation results for the unfaulted case described using a regular $60 \times 220 \times 35$ Cartesian grid. Both grids are partitioned into coarse blocks of $5 \times 5 \times 5$ fine cells each, giving a total of 3696 degrees of freedom for the coarse system, giving an upscaling factor of more than one thousand.

Looking at the error shown in Table 1, we observe that the model run with no fault and homogeneous permeability leads to machine-precision quality of the solution. This is to be expected seeing as the linear flow should be captured exactly by the basis functions; the localization assumption is not an approximation in this case. When adding a fault modeled by corner-point grid with sloping pillars, the error is still fairly low, but the pressure field is unsurprisingly no longer exact. However, the fact that the error is low indicates that the coarse grids should be applicable for more complex permeability fields.

Pressure solutions computed by the MsFV and the fine-scale TPFA methods for the Tarbert layers can be seen in Figure 10. The plots indicate that the MsFV method captures the fine-scale pressure qualitatively very well. However, the discrepancy from the fine-scale solution is significantly higher then for the homogeneous case because complex permeability makes the localization assumptions in the MsFV method weaker. The faulted case sees yet another increase in the errors, mainly because of sharp discontinuities in permeability near the fault. Still, the pressure solution is qualitatively fairly good, with errors in the range to be expected from a non-iterative multiscale method. Since the flux field is constructed via special flux basis functions to get conservative flow, the pointwise flux error is generally expected to be higher than the error in pressure, as is confirmed in Table 1. However, as we will see in the next example, the flux errors can be significantly reduced by updating the correction functions after the first pressure solution.

The Upper Ness formation of SPE10 is very challenging for the MsFV method as the permeability is sharply heterogeneous even at small scales, and high error areas have been reported previously for simulations without iterations on 2D layers (Hajibeygi and Jenny 2011). The problem is aggravated in 3D and Figure 11 shows that the resolution of the MsFV method is quite poor even for the unfaulted case. Adding a displacement along a sloping fault increases the difficulty even further because the fault may introduce sharp changes in permeability due to shifted columns. Figure 11 shows that the solution quality simply is too bad to be considered for any real purposes, even as a first guess in an iterative framework. Improving the MsFV method to handle highly challenging permeability fields the topic for ongoing research, which will likely include updates both to the method and the griding strategies.

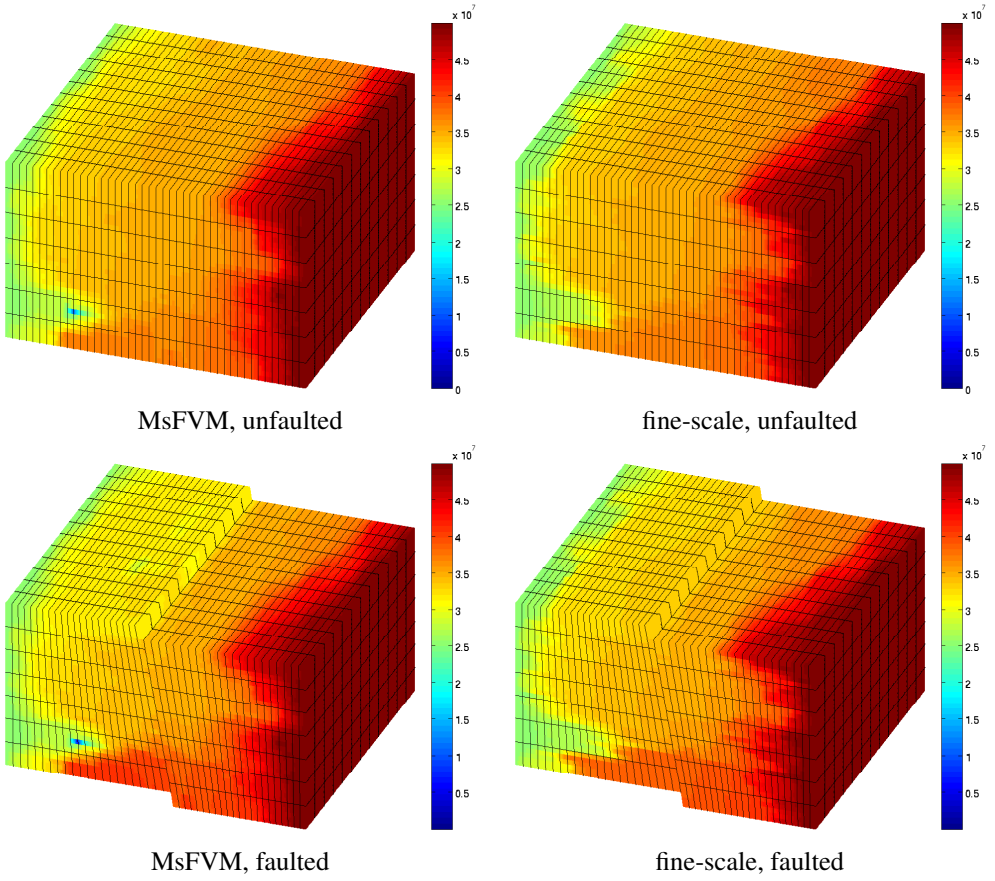


Fig. 10— Pressure in units Pascal computed for the Tarbert formation from the SPE10 data set on the original Cartesian geometry (top) and on the single-fault model from Figure 8 (bottom). Coarse blocks outlined in black.

Preconditioned GMRES on fault. The MsFV method can be used as a preconditioner for iterative solvers to speed up convergence (Lunati et al. 2011). Successive solutions in an iterative solver will try to add in the residual error from the previous solution as source terms for the next multiscale iteration. Using the MsFV method as a preconditioner enables reconstruction of conservative fluxes after any number of iterations, thereby making it possible to converge to some desired precision.

While not the primary focus of this paper, we will briefly demonstrate the MsFV solver's capabilities on a subset of the fault from the previous example. A $30 \times 110 \times 15$ realization is created with permeability sampled from layers of SPE10. Convergence history for both Tarbert and Upper Ness layers can be seen in Figure 12. The GMRES tolerance was set to the default value of 1e-6. After the final iteration, the error is of order 1e-6 for the Upper Ness layers and 1e-9 for the Tarbert layers.

The iterations can also converge for subsets of the more challenging Upper Ness formation. Here, however, the solution quality will be very poor until a large number of iterations have been completed. In areas of large contrasts in permeability, the solution will be patchy and discontinuous until late stages of the iteration process. If we are instead interested in a pressure solution of moderate quality, we can use the permeability field in areas where the method is extra sensitive, namely the face and edge cells, and use the altered permeability field to make a MsFV preconditioner.

It is safe to say that the method is the most sensitive in the first step of the interpolation process, namely at the edges. We will therefore employ a higher degree of smoothing for these cells relative to the face cells, which are less sensitive to contrasts. It is important that the problem is qualitatively close to the original; we do not want to solve another problem entirely.

We begin by selecting a subset of n fine cells that are near the cell where we want to modify the permeability. Weights are assigned to each cell based on the relative proximity to the position of the cell center $\mathbf{x}_0$ in logical space,

$$w_i = 1 - \frac{\|\mathbf{x_0} - \mathbf{x_i}\|_2}{\sum_i \|\mathbf{x_0} - \mathbf{x_i}\|_2}. \tag{12}$$

Once we have the weights, we assign a new permeability value to the fine cell based on the harmonic mean,

$$\tilde{K_0} = \frac{1/n}{\sum_i^n w_i/K_i}. \tag{13}$$

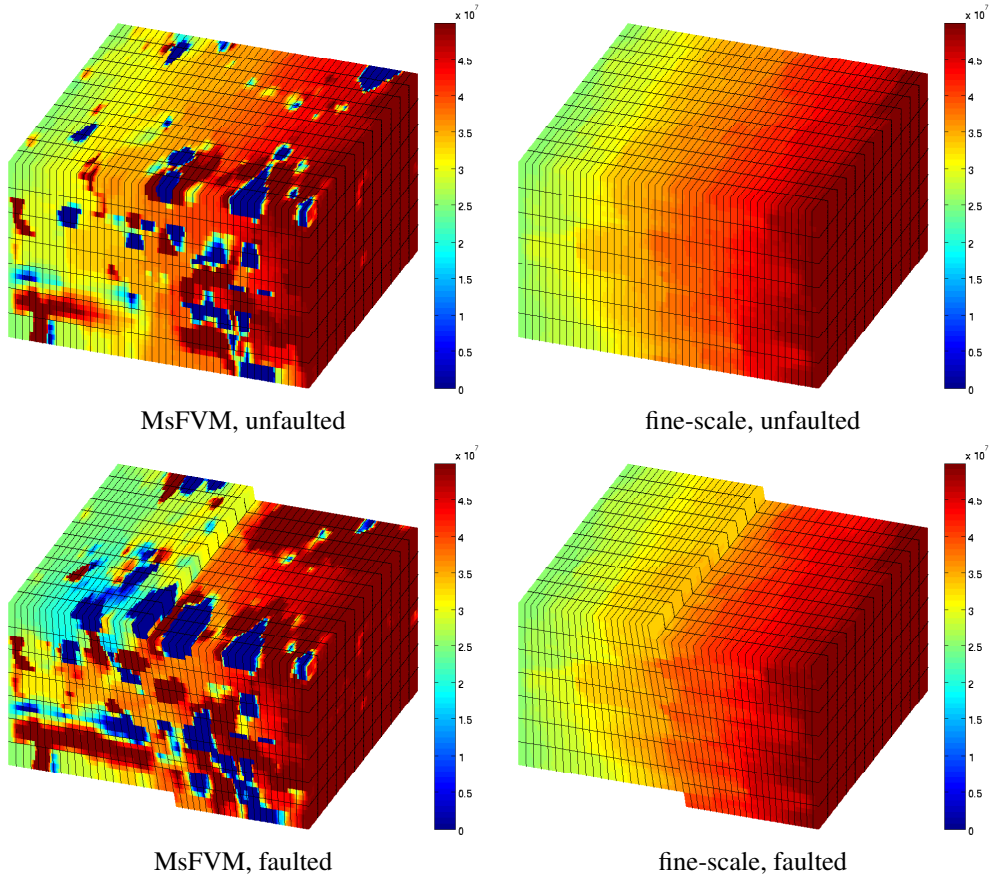


Fig. 11— Pressure in units Pascal computed for the Upper Ness formation from the SPE10 data set on the original Cartesian geometry (top) and on the single-fault model from Figure 8 (bottom). Coarse blocks outlined in black.

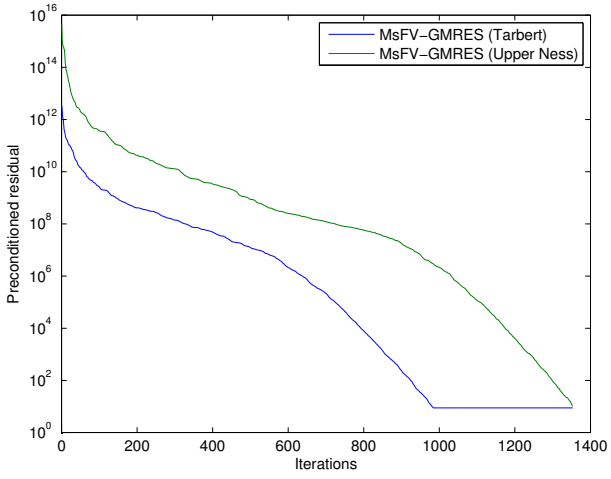
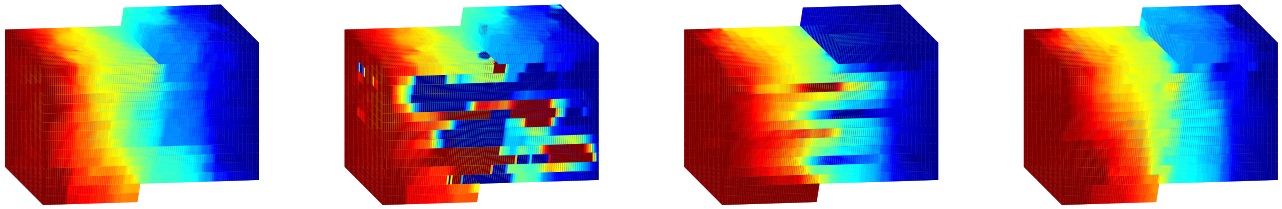


Fig. 12— Convergence history for MsFV-GMRES for Tarbert and Upper Ness layers shown as a plot of the norm of the preconditioned residual as a function the iteration number.

Table 2— Solution disrepancy for three different iterative Multiscale solvers (on the single-fault model from Figure 8) realized on a $30 \times 110 \times 15$ fine grid that is partitioned into a $22 \times 6 \times 3$ coarse grid with permeability from the Upper Ness layers of the SPE10 data set.

Model		Error in pressure		Error in flux	
Permeability	System Matrix	L_2	L_∞	L_2	L_∞
Original	Original	2.60704e+00	7.19786e+01	1.39525e+00	2.91762e+00
Altered	Altered	9.26928e-02	6.14530e-01	9.97036e-01	9.99043e-01
Altered	Original	7.72174e-02	4.43286e-01	9.96027e-01	9.98863e-01



(a) (b) (c) (d)

Fig. 13— Iterative MsFV method using 100 GMRES iterations on the faulted $30 \times 110 \times 15$ Upper Ness model: (a) reference solutions, (b) standard MsFV solution, (c) MsFV solution with altered permeability, and (d) MsFV with altered permeability and original system matrix.

This retains most of the value in relatively homogenous areas, but smooths the permeability in areas of high contrast. For the face and node cells, this is done using cells that are within a logical distance of 5 cells. For the edge cells we use a logical distance of 3 cells, whereas the inner cells are left untouched.

The transmissibilities resulting from these changes in permeability are then used for the multiscale solutions in a preconditioned GMRES solve with 100 iterations. We retain the original system matrix for calculation of residuals to ensure that we are still solving the correct fine-scale system. This will obviously converge slower as the system nears machine precision, but will give much better solutions for the first iterations when solving highly challenging petrophysical cases. The final solutions are plotted in Figure 13. The error can be seen in Table 2 and the result is significant: The error is reduced by a factor of 100 when using MsFV with altered permeability is used as preconditioner for the original system matrix.

Since it is easy to detect where the error is large after an initial solve, a possible preconditioner employing this strategy could smooth the permeability only in certain areas for the first iterations. Since this results in a smoother residual, the rest of the iterations could be done with the original transmissibilities, accelerating convergence in high contrast areas.

A curved fault. Reservoir simulations on realistic grids provide much more challenging structures than the simple fault presented in the previous section. For instance, cells can often be general polyhedral rather than hexahedral and there need not be any logical ordering for the fine cells. This is much more challenging than simple Cartesian cases, for which a dual grid can be constructed simply by exploiting the inherit cell ordering.

To demonstrate the capabilities of the solver on such grids, we will first look at another synthetic case with a vertical fault that describes an S-curve in the horizontal plane. To represent the fault, we will use two different grids. The first grid was created by fitting a triangular grid to the S-curve in the horizontal plane and then extruding it to a prismatic grid in 3D. The second grid was created by first constructing the Voronoi diagram for the areal triangulation and then extruding the resulting polygonal grid to a 2.5D polyhedral PEBI grid. Coarse partitions with approximately 100 coarse blocks for both grids were created using METIS with two layers in the vertical direction for the PEBI grid and one for the prismatic grid. Again, any blocks intersecting the fault were split and blocks with very few fine cells (< 50) were merged with the nearest neighbor. The resulting coarse grids exhibit no Cartesian features in neither of the cases; this can be seen in Figure 14. As in the previous example, flow was driven from left to right by a pressure drop imposed as Dirichlet boundary conditions, here at 1000 and 250 bar respectively. The permeability was assumed to be homogeneous and isotropic.

On the prismatic grid, the multiscale solution is almost indistinguishable from the fine-scale solution. The PEBI grid gives somewhat worse results, which may either be attributed to grid-orientation effects in the coarse-grid stencil or to the fact that there are two coarse blocks in the z-direction as partitioned by METIS. Regardless of the cause, discrepancies in the pressure solution are to be expected near the edges of the primal blocks as this is where the localization assumption may be incorrect.

To reduce the grid-orientation effects on the coarse grid and the impact of the localization assumptions inherit in the method, we can use an inexpensive smoother, e.g., Dirichlet Multiplicative Schwarz (DMS), to remove the corresponding errors and use the resulting difference in the solution as source terms for a new MsFV solution using the existing basis functions. This operation

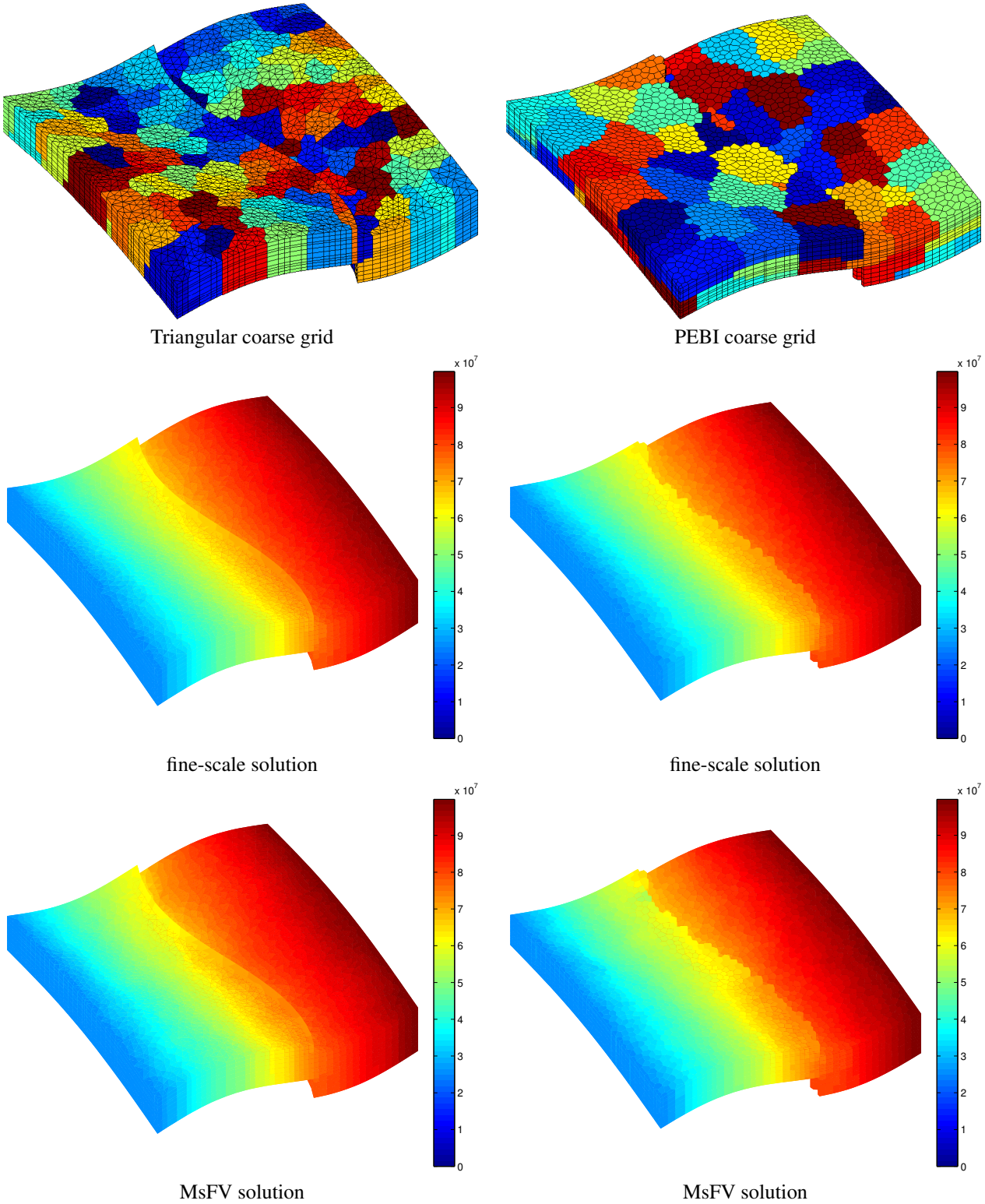


Fig. 14— Prismatic and PEBI grids describing a curved fault (top) along with pressure in units Pascal computed by a TPFA method on the fine grid (middle) and by the MsFV method on a grid with approximately 100 coarse blocks (bottom).

Table 3— Discrepancies between the fine-scale and the MsFV solution computed for two different grids describing a curved fault.

Setup		Error in pressure		Error in flux	
Cell type	Smoothed	L_2	L_∞	L_2	L_∞
PEBI	No	1.35739e-02	7.60413e-02	4.92616e-01	1.40571e+00
PEBI	5 DMS	9.41741e-03	2.53565e-02	1.04888e-01	1.56349e-01
Prismatic	No	1.03102e-02	3.60725e-02	2.86574e-01	8.73046e-01
Prismatic	5 DMS	6.45260e-03	1.36382e-02	7.50301e-02	1.20983e-01

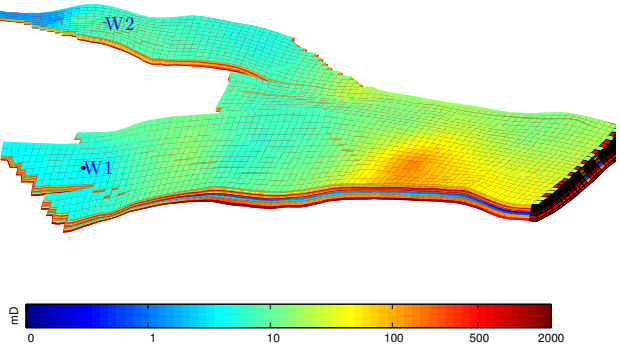


Fig. 15— Horizontal permeability and boundary conditions for the simulation model from the Norwegian Sea. Dirichlet boundary condition is imposed on one end of the reservoir (shown in red), while two producers (W1 and W2) are kept at fixed bottom-hole pressure at the other end of the model.

has a low computational cost, and with only five smoothing cycles the new pressure field is indistinguishable from the true pressure in the visual norm. However, looking at the error in Table 3 it is apparent that the smoother cycle only gives a slight improvement of the L_2 pressure error, but reduces the flux errors as well as the maximum pressure error. Since smoothers in general converge very slowly, they should not be employed uncritically, but a small number of iterations are very beneficial here because they remove oscillatory error terms, giving significantly reduced flux error.

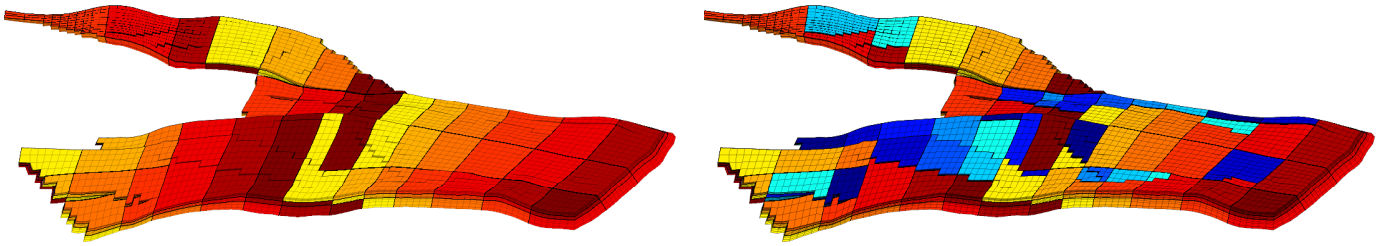
A field model. In the last example, we will use the corner-point grid from the simulation model of an oil reservoir from the Norwegian Sea along with a set of realistic petrophysical properties, as shown in Figure 15. Compared with the previous two examples, the simulation model has a significantly more complex geometry with erosions, pinch-outs, faults, and inactive cells. For instance, while 75% of the fine cells have six faces as in a Cartesian grid, the number of faces per fine cell varies allover from as little as 4 to as much as 19 because of several faults and pinched layers.

To obtain a coarse grid, we will first divide the model logically into coarse blocks as shown in the left part of Figure 16. The initial partition could also have been done geometrically based on cell centroids if logical indexing was not available. To ensure that coarse blocks are split over faults as shown in the right plot of Figure 16, any connections across faults are temporarily removed, and the grid is post-processed to ensure that all cells within each coarse block are connected to all the other cells within the same block.

Two producers operating at fixed bottom-hole pressure $p_{bhp} = 100$ bar are added to the reservoir along with a 500 bar pressure boundary condition at one of the outer boundaries at the opposite side of the wells, as can be seen along with permeability in Figure 15. This gives flow across the entire domain.

We will first compute approximate solutions for uniform permeability equal to 100 mD to ensure that we can judge the quality of the coarse grids separately from the behavior of the method itself on the petrophysical parameters. For uniform permeability (Figure 17), the multiscale solution is approximately equal to the fine-scale solution, aside from some small areas where the localization assumption reduces the accuracy of the method. This indicates that the dual grid has good quality in terms of giving a coarse pressure system and using the result to find fine-scale pressure.

The next step is to solve the same system with realistic permeability. The permeability is layered, inhomogeneous, and anisotropic with six orders of magnitude difference between the minimum and maximum values. The resulting initial MsFV solution (Figure 17) seems to overestimate the pressure in the lower high-permeability layers. It should also be noted that the



Initial partition After splitting by faults

Fig. 16— The corner-point model is partitioned into blocks using the logical IJK cell numbering and then post-processed to create new coarse blocks along the faults.

Table 4— Discrepancies between fine-scale and MsFV solutions on the real-field model.

Setup		Error in pressure		Error in flux	
Permeability	Smoothed	L_2	L_∞	L_2	L_∞
Homogeneous	No	6.80897e-02	7.86341e-01	2.65688e+01	5.01178e+01
Homogeneous	10 DMS	5.52130e-02	2.62088e-01	4.02353e+00	6.98880e+00
Realistic	No	8.87627e-02	6.03834e-01	5.29446e+00	7.91485e+00
Realistic	10 DMS	7.05802e-02	1.70915e-01	1.04733e+00	8.59376e-01

pressure is overestimated near one of the producers, which can be attributed to the fact that the MsFV method treats wells for the coarse system in an integral sense. As before, we can perform a series of inexpensive DMS iterations to produce additional source terms for a better multiscale solution: After ten smoothing steps the resulting multiscale solution is visually much closer to the reference and the flux error reported in Table 4 has been reduced by a factor ten.

Concluding Remarks

We have presented a new formulation of the MsFV method and an accompanying algorithm for creating a coarse dual grid from a general coarse grid, which together enables use of the MsFV method on corner-point grids with faults, erosions, pinch-outs, inactive cells, etc, as well as on and unstructured polyhedral grids. The method was implemented as an open-source prototype solver in MRST, which was then applied to: (i) a simple fault model with complex petrophysical properties described on the corner-point format, (ii) 3D unstructured prismatic and PEBI grids, and (iii) the simulation grid from of a field model from the Norwegian Sea. Judging from our numerical experiments, of which three were reported herein, the errors of the initial MsFV pressure solution are in most cases comparable to what has been reported previously on Cartesian grids with uniform coarse partitions. This leads us to believe that our algorithm is a good starting point to making a robust extension of the MsFV method to 3D stratigraphic and unstructured grids with complex geometry and topology. However, one can easily construct cases for which the current MsFV method produces solutions that deviate strongly from the fine-scale solution. This occurs in particular for scenarios where the permeability field has strong low-high-low or high-low-high variations along the coarse grid faces, which effectively render localization assumption strongly erroneous. Further work should therefore include systematic tests of various grid coarsening strategies, studies of how different degenerate cell geometries will effect the solution quality, as well as development of improved localizations and/or coarsening approaches that adapt to the permeability field to avoid undesired effects of the localizations.

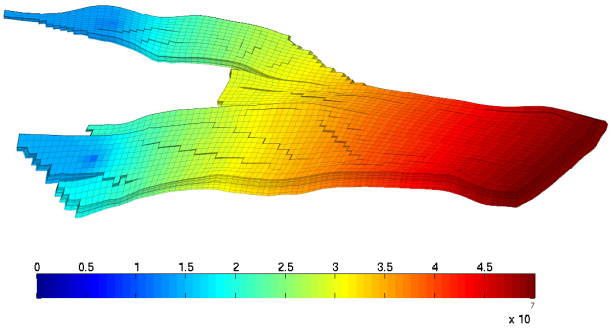
Acknowledgments

The authors thank Bård Skaflestad, Halvor Møll Nilsen, Jostein R. Natvig, and Stein Krogstad for helpful discussions.

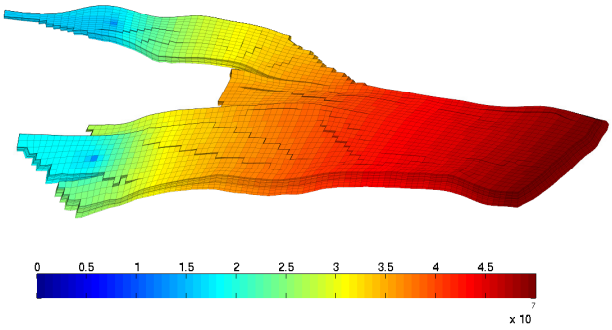
References

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2006. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, 5(2):337–363 (electronic).

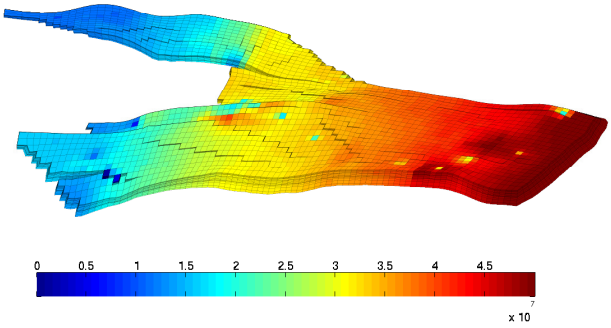
Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2008. Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.*, 12(3):297–315.



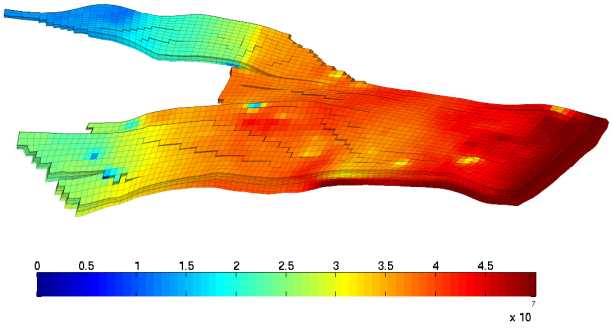
Fine-scale solution, homogeneous K



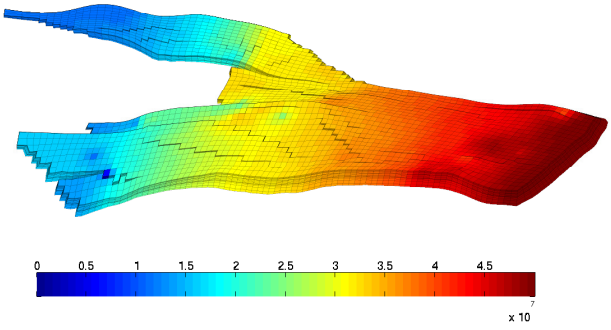
fine-scale solution, inhomogeneous K



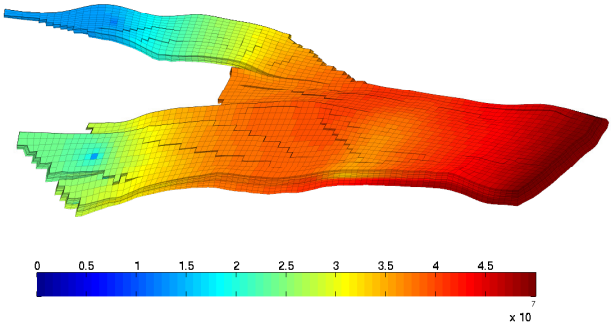
MsFV solution



MsFV solution



MsFV solution + 1 DMS smoothing cycle



MsFV solution + 1 DMS smoothing cyle

Fig. 17— Fine-grid and multiscale solutions computed on the field model with homogeneous permeability (left) and realistic permeability distribution (right).

Alpak, F. O., Pal, M., and Lie, K.-A. 2011. A multiscale method for modeling flow in stratigraphically complex reservoirs. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 21–23 February 2011*. SPE 140403-MS.

Christie, M. A. and Blunt, M. J. 2001. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, 4:308–317. Url: `http://www.spe.org/csp/`.

Efendiev, Y. and Hou, T. Y. 2009. *Multiscale Finite Element Methods*, volume 4 of *Surveys and Tutorials in the Applied Mathematical Sciences*. Springer Verlag.

Hajibeygi, H., Deb, R., and Jenny, P. 2011. Multiscale finite volume method for non-conformal coarse grids arising from faulted porous media. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 21–23 February 2011*. SPE 142205-MS.

Hajibeygi, H. and Jenny, P. 2011. Adaptive iterative multiscale finite volume method. *J. Comput. Phys*, 230(3):628–643.

Hauge, V. L. 2010. *Multiscale Methods and Flow-based Gridding for Flow and Transport in Porous Media*. PhD thesis, Norwegian University of Science and Technology.

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

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

Kippe, V., Aarnes, J. E., and Lie, K.-A. 2008. A comparison of multiscale methods for elliptic problems in porous media flow. *Comput. Geosci.*, 12(3):377–398.

Lie, K., Krogstad, S., Ligaarden, I., Natvig, J., Nilsen, H., and Skaflestad, B. 2012. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16:297–322.

Lunati, I. and Jenny, P. 2006. Multiscale finite-volume method for compressible multiphase flow in porous media. *J. Comput. Phys.*, 216(2):616–636.

Lunati, I. and Jenny, P. 2008. Multiscale finite-volume method for density-driven flow in porous media. *Comput. Geosci.*, 12(3):337–350.

Lunati, I. and Lee, S. H. 2009. An operator formulation of the multiscale finite-volume method with correction function. *Multiscale modeling & simulation*, 8(1):96.

Lunati, I., Tyagi., M., and Lee, S. H. 2011. An iterative multiscale finite volume algorithm converging to the exact solution. *J. Comput. Phys.*, 230(5):1849–1864.

Møyner, O. 2012. Multiscale finite-volume methods on unstructured grids. Master's thesis, Norwegian University of Science and Technology.

MRST 2012. The MATLAB Reservoir Simulation Toolbox, version 2012b. `http://www.sintef.no/MRST/`.

Natvig, J. R., Lie, K.-A., Krogstad, S., Yang, Y., and Wu, X.-H. 2012. Grid adaption for upscaling and multiscale methods. In *Proceedings of ECMOR XIII–13th European Conference on the Mathematics of Oil Recovery*, Biarritz, France. EAGE.

Natvig, J. R., Skaflestad, B., Bratvedt, F., Bratvedt, K., Lie, K.-A., Laptev, V., and Khataniar, S. K. 2011. Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs. *SPE J.*, 16(4).

Nordbotten, J. M. and Bjørstad, P. 2008. On the relationship between the multiscale finite-volume method and domain decomposition preconditioners. *Comput. Geosci.*, 12(3):367–376.

Wang, Y., Hajibeygi, H., and Tchelepi, H. 2012. Algebraic multiscale linear solver for heterogeneous elliptic problems. In *ECMOR XIII - 13th European Conference on the Mathematics of Oil Recovery*, Biarritz, France. EAGE.

Zhou, H. and Tchelepi, H. 2008. Operator-based multiscale method for compressible flow. *SPE J.*, 13(2):267–273.