Doctoral theses at NTNU, 2016:374

Olav Møyner

Next Generation Multiscale Methods for Reservoir Simulation

NTNU
Norwegian University of
Science and Technology

Olav Møyner

Doctoral Thesis

Doctoral theses at NTNU, 2016:374

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Mathematical Sciences

NTNU

ISBN 978-82-326-2092-0 (printed version)
ISBN 978-82-326-2093-7 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Olav Møyner

Next Generation Multiscale Methods for Reservoir Simulation

Thesis for the degree of Philosophiae Doctor

Trondheim, September 2016

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Mathematical Sciences

ISBN 978-82-326-2092-0 (printed version)
ISBN 978-82-326-2093-7 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2016:374

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Abstract

In this thesis, we discuss multiscale simulation techniques for flow and transport in porous media, discretized on unstructured grids. The model equations considered are taken from the field of reservoir simulation, where several orders of magnitude variation in parameters and complex fluid physics result in highly challenging simulation problems.

We consider several novel multiscale solvers and the subsequent extension to compressible multiphase black-oil and compositional flow. The examples include both conceptual models and full field-scale models with industrial complexity in properties, fluid physics, geology and grid types. The thesis also describes the application of numerical experiments termed *flow diagnostics* for optimization and ranking problems, with and without multiscale solvers.

In order to facilitate the research on linear and nonlinear solvers, an open source general purpose simulator framework based on object orientation is developed. This framework aims to enable researchers to rapidly extend or improve existing models and solvers, which is exemplified in this thesis by the development of a simulator for polymer flooding with non-Newtonian fluid physics.

Finally, we also consider transport solvers based on a novel parametrization of Newton updates for the three-phase saturation equations with nonconvex, non-monotone flux functions that can converge for much larger timesteps than the current state of the art solvers.

Contents

Part I: Background

Part II: Scientific Papers

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (Ph.D) at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

Acknowledgments

At the end of my three years of Ph.D. studies, there are many people who should be recognized for their support. I would like to thank my three advisors: Prof. Knut-Andreas Lie at SINTEF, who gave me the opportunity and encouragement to pursue a Ph.D., and for the many productive discussions and writing sessions that resulted in some of the central works in this thesis. Prof. Helge Holden at NTNU was always supportive and helpful during my residency. Jostein Natvig at Schlumberger, for the many discussions about multiscale methods, simulator technology and industrial relevance.

I would also like to thank all the people at SINTEF, NTNU, TU Delft, Schlumberger and Stanford with whom I have co-authored papers. Their contributions body of research in this thesis were invaluable. I am grateful to Prof. Hamdi Tchelepi for hosting me at Stanford for an inspiring research visit from October 2015 - March 2016 that brought new perspectives to my work. My many coworkers at SINTEF also deserve thanks, for the endless number of coffee breaks and many discussions. The financial support provided by the Research Council of Norway under grant no. 226035 and Schlumberger Information Solutions is recognized and greatly appreciated.

I would like to thank my family and friends for their support over the years. Last but not least, I would like to thank my Emily, for her infinite patience and support – especially at times when the research duties infiltrated the household, and displaced other plans.

Olav Møyner
Oslo, September 2016

List of Papers

List of papers included in thesis

I: **The Multiscale Finite Volume Method on Stratigraphic Grids**
Olav Møyner and Knut-Andreas Lie
SPE Journal, volume 19, issue 5, pp. 816-831, 2014
DOI: 10.2118/163649-PA

II: **The Application of Flow Diagnostics for Reservoir Management**
Olav Møyner, Stein Krogstad and Knut-Andreas Lie
SPE Journal, volume 20, issue 2, pp. 306–323, 2014
DOI: 10.2118/171557-PA

III: **A Multiscale Two-Point Flux-Approximation Method**
Olav Møyner and Knut-Andreas Lie
Journal of Computational Physics, volume 275, pp. 273–293, 2014
DOI: 10.1016/j.jcp.2014.07.003

IV: **Construction of Multiscale Preconditioners on Stratigraphic Grids**
Olav Møyner
In proceedings of the 14th European Conference on the Mathematics of Oil Recovery, 2014, Catania, Sicily, Italy.
DOI: 10.3997/2214-4609.20141775

V: **A Multiscale Restriction-Smoothed Basis Method for High Contrast Porous Media Represented on Unstructured Grids**
Olav Møyner and Knut-Andreas Lie
Journal of Computational Physics, volume 304, pp. 46-71, 2016
DOI: 10.1016/j.jcp.2014.07.003

VI: **Application of Flow Diagnostics and Multiscale Methods for Reservoir Management**

Knut-Andreas Lie, Olav Møyner and Stein Krogstad
In proceedings of the 2015 SPE Reservoir Simulation Symposium, Houston, Texas, USA
DOI: 10.2118/173306-MS

VII: **MRST-AD – An Open-Source Framework for Rapid Prototyping and Evaluation of Reservoir Simulation Problems**
Stein Krogstad, Knut–Andreas Lie, Olav Møyner, Halvor Møll Nilsen, Xavier Raynaud and Bård Skaflestad
In proceedings of the 2015 SPE Reservoir Simulation Symposium, Houston, Texas, USA
DOI: 10.2118/173317-MS

VIII: **A Multiscale Restriction-Smoothed Basis Method for Compressible Black-Oil Models**
Olav Møyner and Knut-Andreas Lie
SPE Journal, published ahead of print, June 2016
DOI: 10.2118/173265-PA

IX: **Multiscale Simulation of Polymer Flooding with Shear Effects**
Sindre Hilden, Olav Møyner, Knut-Andreas Lie and Kai Bao
Transport in Porous Media, volume 113, issue 1, pp. 111–135, 2016
DOI: 10.1007/s11242-016-0682-2

X: **The Multiscale Restriction Smoothed Basis Method for Fractured Porous Media (F-MsRSB)**
Swej Shah, Olav Møyner, Matei Tene, Knut-Andreas Lie and Hadi Hajibeygi
Journal of Computational Physics, volume 318, pp. 36–57, 2016
DOI: 10.1016/j.jcp.2016.05.001

XI: **Fully Implicit Simulation of Polymer Flooding with MRST**
Kai Bao, Knut-Andreas Lie, Olav Møyner and Ming Liu
In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601880

XII: **Successful Application of Multiscale Methods in a Real Reservoir Simulator Environment**
Knut-Andreas Lie, Olav Møyner, Jostein Roald Natvig, Antonina Kozlova, Kyrre Bratvedt, Shingo Watanabe and Zhouyi Li
In proceedings of the 15th European Conference on the Mathematics of

Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601893

XIII: **Nonlinear Solver for Three-phase Transport Problems Based on Approximate Trust Regions**
Olav Møyner
In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601899

XIV: **A Multiscale Restriction-Smoothed Basis Method for Compositional Models**
Olav Møyner and Hamdi Tchelepi
Accepted for the proceedings of the SPE Reservoir Simulation Conference 2017, Woodlands, Texas, USA

Papers not included in thesis

Here follows a list of papers not included in the thesis. For papers I and II, these are conference procedings that were later reworked into journal publications that are included in the thesis. Papers III, IV and V concern work on CO_2 storage outside the scope of this thesis.

I: **The Multiscale Finite Volume Method on Unstructured Grids**
Olav Møyner and Knut-Andreas Lie
In proceedings of the 2013 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA
DOI: 10.2118/163649-MS

II: **A Multiscale Method Based on Restriction-Smoothed Basis Functions Suitable for General Grids in High Contrast Media**
Olav Møyner and Knut-Andreas Lie
In proceedings of the 2015 SPE Reservoir Simulation Symposium, Houston, Texas, USA
DOI: 10.2118/173265-MS

III: **Spill-Point Analysis and Structural Trapping Capacity in Saline Aquifers Using MRST-co2lab**
Halvor Møll Nilsen, Knut-Andreas Lie, Olav Møyner and Odd Andersen
Computers & Geosciences, volume 75, pp. 33–43, 2015
DOI: 10.1016/j.cageo.2014.11.002

IV: **A Simulation Workflow for Large-scale CO2 Storage in the Norwegian North Sea**
Knut-Andreas Lie, Halvor Møll Nilsen, Odd Andersen and Olav Møyner
Computational Geosciences, volume 20, issue 3, pp. 607–622, 2016
DOI: 10.1007/s10596-015-9487-6

V: **A Simulation Workflow for Large-scale CO2 Storage in the Norwegian North Sea**
Knut-Andreas Lie, Halvor Møll Nilsen, Odd Andersen and Olav Møyner
In proceedings of the 14th European Conference on the Mathematics of Oil Recovery, 2014, Catania, Sicily, Italy
DOI: 10.3997/2214-4609.20141877

Part I

Background

Chapter 1

Introduction

The simulation of flow and transport in porous media is important for many industrial and scientific applications. The equations used to model creeping, potential-driven flow in pore networks are applicable for a wide range of large-scale industrial processes, including different stages of hydrocarbon recovery, subsurface sequestration of greenhouse gases, geothermal energy, groundwater management and contamination studies for nuclear or toxic materials. Small-scale processes are also governed by the same equations, including aqeuous infiltration into concrete, the design of passive filters and kidney function.

For many practical applications, numerical simulation is necessary to evaluate the possible outcomes for any given scenario. In this thesis, we consider the numerical simulation of flow and transport in heterogeneous reservoirs with an emphasis on processes relevant for secondary and tertiary recovery of hydrocarbons. If we take the oil and gas resources located on the Norwegian continental shelf as a motivating example, the reservoirs are located off-shore and at depths of several kilometers below sea level [94]. The extreme conditions of these resources make extraction costs significantly higher than for example shallower reservoirs located on land in the Middle-East. As a result of high cost, each engineering decision must be carefully evaluated while accounting for uncertainty in geology, fluid behavior and hydrocarbon distribution. Reservoir simulation is an integral part of this process, where better simulation techniques increase both the accuracy of the predictions and the number of different scenarios that can be evaluated in a reasonable time frame. Faster, more accurate simulators can thereby contribute extend the lifetime of mature Norwegian oil fields and assist in the complex late-stage recovery process where enhanced oil recovery is deployed.

Mathematically, the governing equations for porous media flow on a geomodel scale are interesting because they have large variations in medium properties on both very short and very long scales. Consequently, the definition of a scale for a meaningful representative elementary volume (REV) where the properties can be homogenized without incurring severe errors is challenging. In addition, the thermodynamic behavior of the fluid is highly nonlinear, indicating that the physical mechanisms of oil recovery can be very challenging to simulate accurately.

In the decades since the first computers were made available for scientific use, the computing landscape has undergone an revolution in capabilities that has taken reservoir simulators from models with one-dimensional single-phase flow with ten or so cells to fully three-dimensional multicomponent models with hundreds of millions of fine cells [26]. However, as the transistor density in CPUs approaches the theoretical limits imposed by the laws of physics, computer architectures have transitioned to parallel, distributed computing machines for which algorithms designed for serial processors may no longer be relevant. The next generation of reservoir simulation requires new algorithms that scale for such highly parallel computers. These algorithms must correctly decouple the different physical effects to scale for problem sizes beyond what is currently possible.

Multiscale methods is one proposed technology which aims to correctly resolve the complex interplay of local flow on small length scales and global flow on long time scales using a reduced number of unknowns (degrees of freedom) associated with the coarse scales for highly heterogeneous reservoirs. The main goal of this thesis is to further develop multiscale methods to a level where they are applicable to the physics, grids and strong parameter heterogenity relevant for modern reservoir simulation.

1.1 Outline of this thesis

This thesis is divided into two main parts. The first part is a general introduction to the relevant physics, model equations and discretization techniques under consideration. Part I, Chapter 2 gives an introduction to the different grids, physics and governing equations used in Part II, illustrated by a few examples that use the Matlab Reservoir Simulation Toolbox (MRST), an open source platform for research on numerical methods for porous media. Chapter 3 details the discretizations and different techniques used to solve the model equations from Chapter 2. Chapter 4 gives a brief introduction to multiscale solvers for porous media flow. In Chapter 5, a description of the object-oriented framework used to implement and simulate the results in the papers is given. Chapter 6 describes and hopefully motivates each of the fourteen papers included in Part II. Finally, Chapter 7 contains concluding remarks, as well as some indications to possible future research directions.

Part II contains the scientific papers that make up the bulk of the thesis. The order of papers is semi-chronological, where some entries have been rearranged for benefit of the reader. Five additional papers are listed, that were not included in the thesis, but never-the-less were written during the doctoral scholarship.

Chapter 2

Flow and Transport in Porous Media

2.1 Grids in subsurface models

In many idealized numerical applications, the grid used to discretize the domain is not given special care. Often, the model equations are solved on the unit square where the grid can be freely chosen for accuracy or convenience as the nature of the equation does not vary spatially. This is not the case in reservoir simulation, and the grids used have several peculiar features, due to the physical properties they are intended to model. Grids representing subsurface reservoirs can contain cells with very large aspect ratios. For instance, a typical reservoir has typically a vertical thickness of at most a few hundred meters, corresponding to strata that may have been deposited over geological timescales of several million years. At the same time, the reservoir can span ten or even hundreds of kilometers horizontally, and compartments only a few meters apart vertically can be sealed off from fluid exchange with each other, while still being in communication with other areas hundreds or thousands of meters away in the lateral direction. As a result, the grids often have very thin cells that gives high vertical resolution to account for geological layering. Figure 2.1 is a conceptual drawing of a reservoir model that consists of three different geological layers and three faults.

Layering is not the only feature that makes reservoir grids challenging, however. Geological processes have over millions of years twisted, crushed and deformed the structure of the rocks. Large volumes may be eroded away, and faults and fractures can intersect the neatly layered rock strata to form highly irregular structures. The industry standard corner-point for-

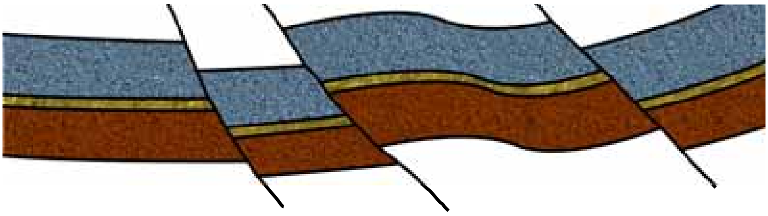


Figure 2.1: Conceptual drawing of a reservoir. Different layers from different geological eras are present along the vertical direction, but other processes has displaced and deformed the layering structure. The dimensions of this figure is not to scale. The impermeable top and bottom layers are transparent.

mat used to create reservoir grids is fundamentally a structured format in which hexahedral cells are stacked logically upon each other to define a mesh of pillars that cover the extent of the reservoir. The cells inside each pillar represent different geological depositions, with the oldest depositions at the bottom and the youngest at the top. This means that cells with the same index in the vertical numbering correspond to rock units deposited at approximately the same time (like the three layers in Figure 2.7). To account for subsequent geological activities, the pillars are translated and skewed, cells are shifted upward or downward to represent faults, or deformed or even removed to model erosion and different degrees of compaction. Hence, the resulting grids after processing will almost always contain highly unstructured features.

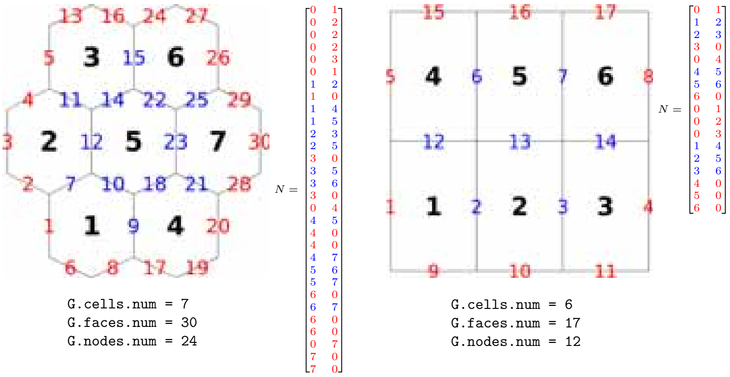
The Matlab Reservoir Simulation Toolbox (MRST)

All methods implemented and subsequent numerical experiments that form the basis of this thesis have been carried out using the Matlab Reservoir Simulation Toolbox (MRST), which is an open-source toolbox for Matlab released by the Computational Geoscience group at SINTEF Applied Mathematics. Originally intended as a platform for implementing consistent discretizations for unstructured grids [66, 67], the current release of MRST includes a large number of different routines for defining, solving and visualizing porous media problems. MRST is used extensively by a large number of researchers and student all over the world. By working with an established platform for research, individual contributions build upon each other, resulting in an end result that is greater than the sum of its individual parts. New releases of MRST are released twice each year, where the individual releases reach between 1000 and 2000 unique downloads each.

Unstructured grids in MRST

MRST was designed to support unstructured grids. For this reason, MRST represents all grids, regardless of type, as unstructured grids internally. The benefit is that all grids are given in a common format and routines will automatically support new grids as long as they can be re-interpreted as unstructured grids. The downside is that simple grids will require more memory to be stored and the code structure can be unfamiliar to users who are used to developing for fully structured grids.

Unstructured grids are chiefly characterized by the lack of clear indexing structure, making it impossible to tell if any two cells are in proximity to each other simply based on the cell numbers. In MRST, we assume that any pair of cells can be connected. All grids are assumed to have n_c grid cells, n_f faces and n_n nodes. The relationship between different cells is defined by the interfaces and consequently each grid contains a list of face connections which is a matrix with $n_f \times 2$ entries. A face i is connected to cells `N(i, 1)` and `N(i, 2)`. If one of the values is a zero, the face is on the boundary and is consequently only connected to a single cell. Figure 2.2 demonstrates the neighborship matrix for two different grids: A fully unstructured Voronoi grid and a regular Cartesian grid. Treating the topology of a grid model as a graph was first introduced to reservoir simulation in Lim [68].



$$N = \begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 3 \\ 0 & 1 \\ 1 & 2 \\ 1 & 0 \\ 1 & 4 \\ 1 & 5 \\ 2 & 3 \\ 2 & 5 \\ 3 & 0 \\ 3 & 5 \\ 3 & 6 \\ 3 & 0 \\ 0 & 4 \\ 4 & 5 \\ 4 & 0 \\ 4 & 0 \\ 4 & 7 \\ 5 & 6 \\ 5 & 7 \\ 6 & 0 \\ 6 & 7 \\ 6 & 0 \\ 6 & 0 \\ 0 & 7 \\ 7 & 0 \\ 7 & 0 \end{bmatrix}$$

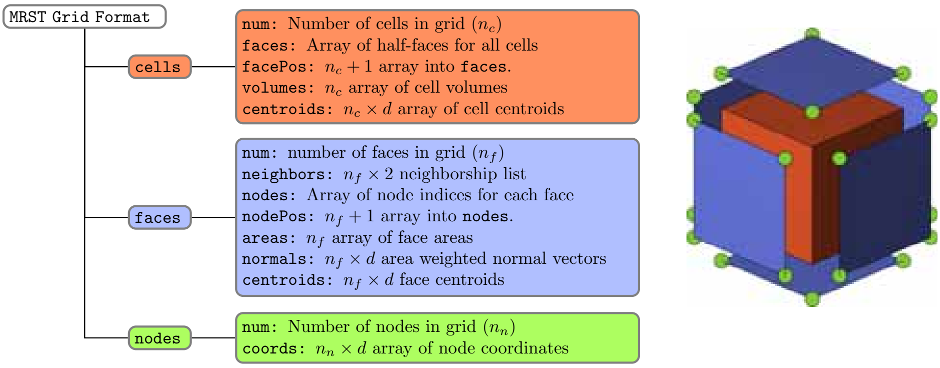
```
G.cells.num = 7
G.faces.num = 30
G.nodes.num = 24
```

$$N = \begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 2 & 3 \\ 3 & 0 \\ 0 & 4 \\ 4 & 5 \\ 5 & 6 \\ 6 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 3 \\ 1 & 4 \\ 2 & 5 \\ 3 & 6 \\ 4 & 0 \\ 5 & 0 \\ 6 & 0 \end{bmatrix}$$

```
G.cells.num = 6
G.faces.num = 17
G.nodes.num = 12
```

Figure 2.2: Two different grids represented in the unstructured MRST format. The Voronoi (left) and Cartesian (right) grids can both be defined in the form of a connectivity matrix for the interfaces.

The fully unstructured philosophy extends to the definition of the primitive properties of the grid. MRST has a constructive approach to the grid data structure, where each cell is defined by a list of the faces that make up its surface. Faces themselves are again made up of nodes, which finally are comprised of actual coordinates that place the nodes, cells and faces spatially, as shown in Figure 2.3. This format does not make any assumptions about structure and any cell can have any number of faces that each themselves can be general polygons. Once the grid has been constructed, it is customary to apply the `computeGeometry` function that accurately determines the cell and face centroids, the face areas and cell volumes and the normal vectors of each face. Geometry information will be required when we later on will discretize our model equations. For detailed description of the grid structure in MRST, the reader is deferred to papers and books that go into more detail [53, 66, 67]. Papers VII, VIII and XI all touch upon the way grids and discrete operators are constructed in MRST.



`num`: Number of cells in grid (n_c)
`faces`: Array of half-faces for all cells
`facePos`: $n_c + 1$ array into `faces`.
`volumes`: n_c array of cell volumes
`centroids`: $n_c \times d$ array of cell centroids

`num`: number of faces in grid (n_f)
`neighbors`: $n_f \times 2$ neighborship list
`nodes`: Array of node indices for each face
`nodePos`: $n_f + 1$ array into `nodes`.
`areas`: n_f array of face areas
`normals`: $n_f \times d$ area weighted normal vectors
`centroids`: $n_f \times d$ face centroids

`num`: Number of nodes in grid (n_n)
`coords`: $n_n \times d$ array of node coordinates

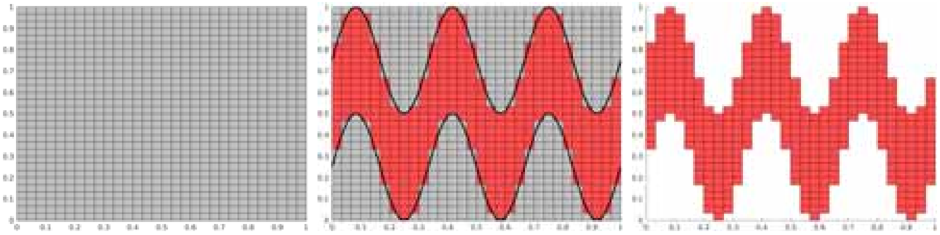
Figure 2.3: Visual breakdown of the unstructured grid format used in MRST: The grid is comprised of cells, faces and nodes, all which come together to form the final grid shown on the left.

Examples of reservoir grids

We will illustrate the variety and complexity of reservoir grids by the way of a few worked MRST examples. We begin by considering a simple Cartesian grid that discretizes the unit square $\Omega = [0,1] \times [0,1]$ into 30×30 cells of equal size as seen in Figure 2.4a.

```
% Define a 25x25 grid discretizing the unit square
G = cartGrid([30, 30], [1, 1]);
% Add geometry information
G = computeGeometry(G);
plotGrid(G, 'FaceColor', [0.267, 1.000, 0.730])
```

A common technique in subsurface modeling is to start with some structured grid of the right dimensions and then remove any cells that either do not



(a) Initial grid (b) Flagged cells (c) Extracted subgrid

Figure 2.4: A Cartesian grid is converted to a cutout grid using bounding functions in order to get a grid with a sine pattern.

exist in the physical domain or are impermeable to flow. Let us assume that we know that the region of interest is bounded by two functions $f(x), g(x)$ that may represent the horizons of the geological layer we want to study,

$$f(x) = \frac{1}{4} + \frac{1}{4}\sin(6\pi x), \qquad g(x) = \frac{3}{4} + \frac{1}{4}\sin(6\pi x).$$

We define these functions and use them to flag and plot the subset of cells where $f(x) < y < g(x)$ in Figure 2.4b,

```
% Get cell x and y centroids
x = G.cells.centroids(:, 1); y = G.cells.centroids(:, 2);
% Define bounding functions, and flag cells that will be kept
fnUnder = @(x) 0.25*sin(x*6*pi) + 0.25;
fnTop = @(x) 0.25*sin(x*6*pi) + 0.75;
ok = y > fnUnder(x) & y < fnTop(x);
hold on
% Plot functions bounding the domain
plot(xs, fnUnder(xs), 'k','linewidth', 2), plot(xs, fnTop(xs), 'k','linewidth', 2)
% Plot the cells that is to be kept
plotGrid(G, ok, 'FaceColor', [1.00, 0.30, 0.30])
% Plot the cells that will be removed
plotGrid(G, ~ok, 'FaceColor', [0.75, 0.75, 0.75])
xs = 0:0.01:1;
plot(xs, fnUnder(xs), 'k','linewidth', 2)
plot(xs, fnTop(xs), 'k','linewidth', 2)
```

Finally, we remove the excess cells and plot the final grid (seen in Figure 2.4c):

```
G_sub = extractSubgrid(G, ok);
plotGrid(G_sub, 'FaceColor', [0.267, 1.000, 0.730])
```

As an example of a realistic grid, we consider the corner-point grid from the Norne field model [41, 109]. Norne is an oil reservoir in the Norwegian

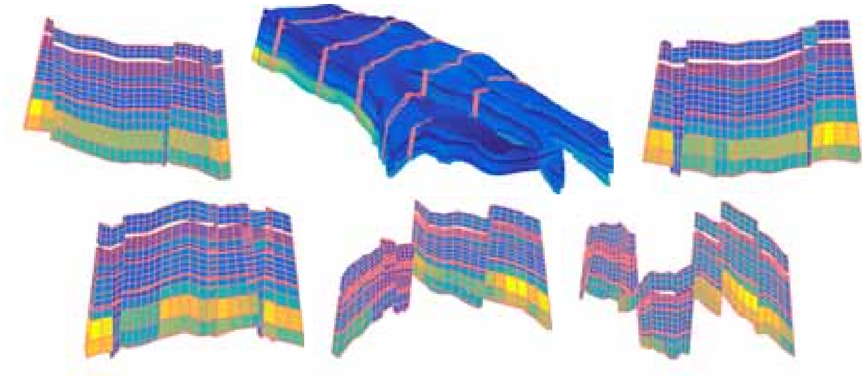


Figure 2.5: Five different cross sections of the cells in the Norne field model, taken perpendicular to the major fault lines. The cells are colorized according to their relative bulk volumes. Note that the vertical axis is exaggerated by a factor 5 for visualization purposes.

Sea, for which the reals imulation model has been released to under the permissive Open Database License [92]. As discussed above, The corner-point format is a common way of specifying models for reservoir simulation. We see that the corner-point format for the Norne model results in cells that are far from ideal hexahedra, as it contains a number of faults and eroded cells, making the final processed grid essentially unstructured in regions. To download, read and process the Norne model in MRST, it is sufficient with seven lines of code:

```
mrstModule add deckformat
makeNorneSubsetAvailable();
makeNorneGRDECL();
grdecl = fullfile(getDatasetPath('norne'), 'NORNE.GRDECL');
grdecl = readGRDECL(grdecl);
grdecl = convertInputUnits(grdecl, getUnitSystem('METRIC'));
G      = processGRDECL(grdecl);
```

The processed reservoir model is plotted in Figure 2.5. To illustrate the locally unstructured features, several cross-sections over the fault lines are included, demonstrating certain highly irregular cells with large aspect ratios and non-neighboring connections. The Norne field model is used as an example in many of the papers of this thesis.

2.2 Petrophysical properties

Permeability and porosity

Once we have obtained the grid structure that describes the geometry and internal structural architecture of the reservoir, we must populate the cells with petrophysical properties that reflect the different rock types (sedimentology) in the model. Since the underlying medium can be highly heterogeneous, we will require the definitions of several properties. The porous medium consists of particulate matter that has been compressed together to form a static rock structure. In between the different grains of sand, there exists a void space where fluid can flow. The fraction of the (geometrical) bulk volume available to fluid flow is called the effective *porosity* (ϕ), which is a dimensionless fraction that signifies how much of the bulk rock volume (V_b) is void space accessible to flow. A conceptual model of a porous medium on the microscale with different grain sizes is shown in Figure 2.6. It is then natural to define the pore *volume* as the actual volume available to fluid flow,

$$\Phi = \phi V_b, \quad 0 \leq \phi \leq 1$$

The porosity only defines the porous medium's ability to store fluids, so we also require a way to define the rate of fluid flow. The *permeability* models the rock's ability to conduct fluid flow and is represented as a symmetric, positive definite tensor $\mathbf{K}$. The permeability can vary greatly between different regions in reservoir models and relates the pressure gradient to the actual flow rates in the medium. When the pore throats have smaller diameters, a higher pressure differential will be required to force the same amount of fluid through. Darcy's law is used to model this relationship and for single-phase flow without gravity or capillary forces it takes the form,

$$\vec{v} = -\mathbf{K}\nabla p,$$

where $\vec{v}$ is the velocity field and p the pressure field. Darcy's law is a reasonable empirical law for creeping flow where fluid momentum has negligible contributions to the flow field. Fluid flowing along the negative potential gradient is analogous to Fourier's law in heat transfer and Fick's law for diffusion problems.

Examples of grids with petrophysical properties

The proceeding section defined permeability and porosity. To better understand the types of values encountered in practice, we will examine a few example datasets that are easily accessible using MRST.

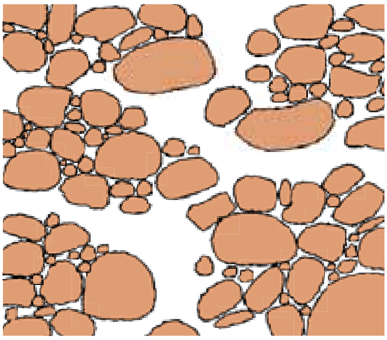


Figure 2.6: Idealized example of a rock cross-section. The white space represents the void space accessible to fluid flow, while the light brown represents the impermeable part of the medium, corresponding to solid mineral grains.

A synthetic layered model

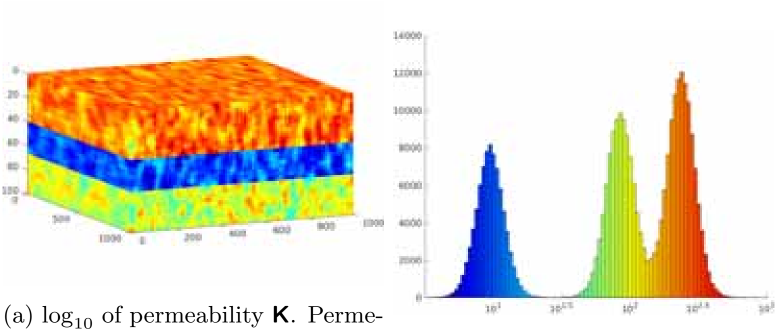
The first model will be a wholly synthetic model that has been chosen to illustrate a typical situation in a reservoir model, where multiple different rock types are arranged in layers according to their geological age. We will consider a 1000 × 1000 × 100 m^3 model, with 100 × 100 × 30 cells. We can quickly set up this model using MRST's built-in routines:

```
% Define a grid with 30,000 cells and 1000x1000x100 meter dimensions
G = cartGrid([100, 100, 30], [1000, 1000, 100]);
% Define three layers with 250, 10 and 90 mD means
K = logNormLayers(G.cartDims, [250, 10, 90]);
% Create rock structure with uniform porosity
rock = makeRock(G, K*milli*darcy(), 0.3);
```

Once we have set up the grid and the corresponding rock structure, we can examine the model visually using standard MRST plotting routines. We plot the permeability as cell-wise data on the grid,

```
% Take the log10 of the dataset
d = log10(rock.perm/(milli*darcy()));
plotCellData(G, d, 'EdgeColor', 'none')
axis tight
view(65, 10)
```

Note that we have taken the base 10 logarithm of the permeability. Taking the logarithm when plotting permeability is common because the underlying data varies greatly and a linear color scale over-emphasizes the extremal values. We also plot a histogram of the distribution, colorized by the same colormap as in the call to `plotCellData`:



(a) $\log_{10}$ of permeability $\mathbf{K}$. Permeability unit is millidarcy.

(b) Histogram of $\log_{10}\mathbf{K}$

Figure 2.7: Permeability data for the synthethic layered model used to demonstrate permeability. Note the three gaussian distributions in the histogram which correspond to each of the three different rock-types in the model.

```
figure;
colorizedHistogram(d, 100)
xlim([min(d), max(d)])
x = get(gca, 'XTick');
xt = arrayfun(@(x) ['10^{', num2str(x), '}'], x, 'UniformOutput', false);
set(gca, 'XTickLabels', xt)
```

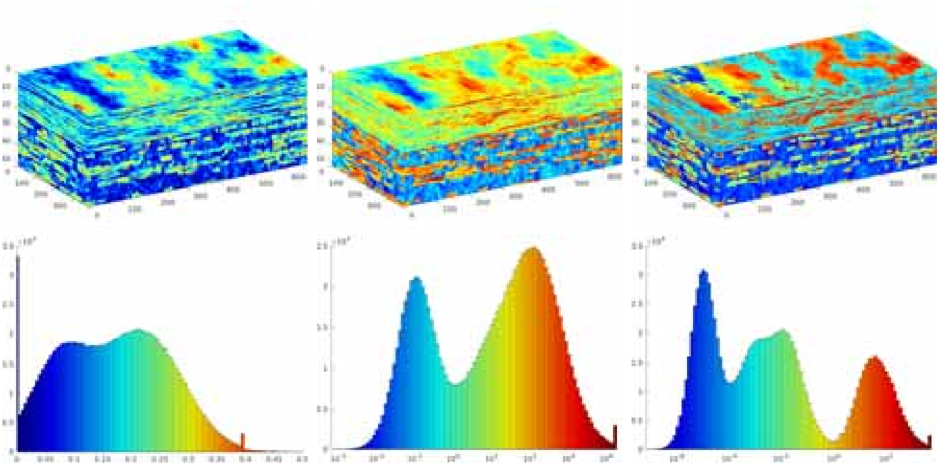
Figure 2.7 contains the resulting plots. We immediately see that there are three very different rock-types based on the color grouping. If we look at the histogram of the permeability distribution, there are three different Gaussian distributions. In the call to `logNormLayers` we specified the means as 250, 10 and 90 mD, respectively, and this is also present in the final model. We also see that within each layer there is a log-normal stochastic distribution of the permeabilities. While highly idealized, this is somewhat representative of how simulation models are built. The models are built using a mixture of various data sources that all have different accuracy. Seismic data is used to place the horizons between the layers and core samples from wells give highly detailed point samples of the geological distribution. To obtain probable petrophysical properties in areas without wells, equiprobable statistical realizations are generated and used in a *history matching* process where statistical parameters are tuned such that the simulation results match the known production history.

Model 2 from the Tenth SPE Comparative Solution Project

The comparative solution project is a series of papers published by the Society of Petroleum Engineers that compare different simulators and numerical techniques on a wide variety of benchmark datasets. The tenth study [19], in particular, has become a de facto benchmark for new computational methods. The objective of the study itself was to test different upscaling methods, but the second dataset from this study has become very popular among researchers in general, likely due to the geometry being Cartesian and the petrophysical data being freely available.

The grid itself represents a box domain of size 1200 × 2200 × 170 ft^3 with 60 × 220 × 85 cells. The model contains two very different permeability regions. The top 35 layers are sampled from the Tarbert formation and contains a (relatively) smoothly varying permeability field, while the remaining 50 layers are sampled from the Upper Ness formation, which contains a fluival distribution where highly permeable sands form channels that intersect low-permeable mudstone. As the transition between the two rock types is abrupt, the resulting permeability field has extreme transitions in contrast.

The MRST module `spe10` contains the required routines for downloading and displaying the dataset and Figure 2.8 shows the porosity, vertical and lateral permeabilities. We can clearly see how we have in the same dataset several different permeability distributions that overlap in parts, with significant anisotropy as the vertical and lateral permeabilities differ by several orders of magnitude. The maximal variation in permeability is over ten orders of magnitude for the entire model. Note also that the porosity contains a number of cells with zero porosity. Such cells are usually removed in a pre-process step by the simulator, but the zero porosities can also be replaced by small values to allow for simulation.

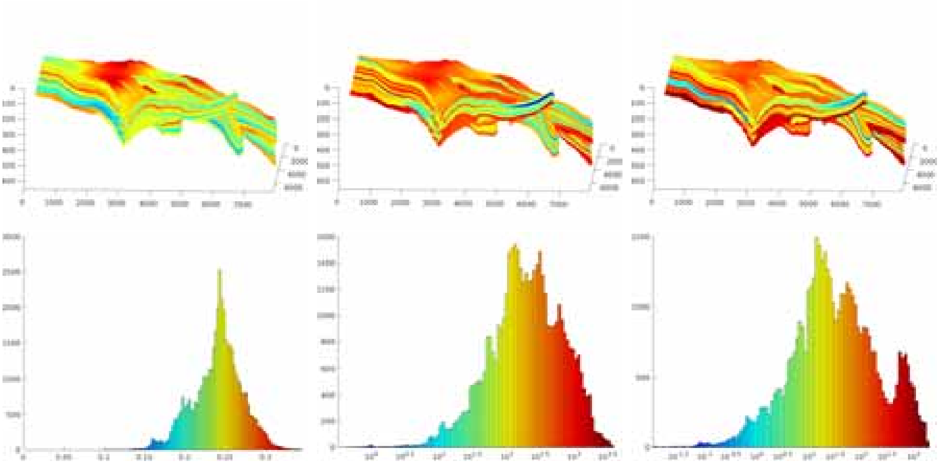


(a) Porosity ϕ (b) $\log_{10}$ of $\mathbf{K}_x$. (c) $\log_{10}$ of $\mathbf{K}_z$.

Figure 2.8: The second model from the SPE10 dataset. Note that the top 35 layers correspond to the Tarbert formation and thus have a different permeability and porosity distribution than the remaining 50 layers which belong to Upper Ness.

The Norne field model

In Section 2.1 we saw that the Norne field had a complex geometrical structure due to geological features. We can now examine the petrophysical propertes of the grid, shown in Figure 2.9. The histograms for vertical and horizontal permeability indicates a highly heterogenous reservoir, with several orders of magnitude variations in permeability. Once plotted directly on the grid cells, we can see that the permeabilities form a layered structure. The many faults disturb this structure, however, leading to discontinuities in the permeability in the lateral direction where different layers intersect. The permeability varies over four orders of magnitude, with significant differences in distribution for the vertical and the lateral direction.



(a) Porosity ϕ (b) $\log_{10}$ of $\mathbf{K}_x$. (c) $\log_{10}$ of $\mathbf{K}_z$.

Figure 2.9: Petrophysical properties of the Norne field model. The layered structure of the permeability leads to a non-trivial histogram and varying anisotropy gives the vertical permeability a differently shaped distribution than the horizontal permeability. As the grid has a vertical extent of 600 meters and a length of 7000 meters, the vertical size of the model has been exaggerated for the purpose of illustration.

2.3 Fundamental equations

We have up to this point described the grids and medium properties used in the simulation of flow and transport in porous media. However, we have not yet described the governing equations and the fluid physics required to pose a simulation problem. Conceptually, these equations are the same for a wide range of physical processes, including the simulation of CO_2 storage, geothermal energy production, ground-water management, filter design and even for *in vivo* processes including kidney function or oxygenation in lung tissue. The model problem for this thesis, however, is the simulation of hydrocarbons production during what is termed the secondary and tertiary recovery stages. The production of hydrocarbons can be divided into three stages. The first stage, primary production, relies on the overburden pressure of the reservoir to force out oil and gas from the production wells. In the secondary stage, injection wells are drilled to force more hydrocarbons out using a combination of water injection and gas-reinjection. The tertiary stage, sometimes referred to as enhanced oil recovery, is a late stage strategy for mature reservoirs where the injected water may be heated or mixed with chemical/biological compounds to improve the recovery of the residual oil.

Most Norwegian-operated reservoirs are at late stages in their lifetime and reservoir simulation is an essential tool to plan new wells, injection rates and EOR-strategies to extend the lifetime and the amount of resources recovered. Indeed, the Norwegian government mandates that a certain percentage of the estimated fluid volumes are to be extracted regardless of economic viability to ensure that natural resources are not wasted.

In Figure 2.10 we can see the same reservoir as in Figure 2.1, but we have now added fluids and wells to the drawing. Initially, the reservoir will be undisturbed and water, oil and gas (if present) will have been segregated according to densities over the aeons. Production wells are drilled to extract the valuable oil and gas resources and eventually, as the reservoir transitions over to secondary production, new wells will be drilled to inject fluids to sweep out remaining hydrocarbons and keep the reservoir pressure from dropping too much. As this happens, the forces that have been in equilibrium for millennia are disturbed and the flow patterns that appear are the result of a balance between viscous, gravitational and capillary forces that operate on different scales, with highly nonlinear behavior.

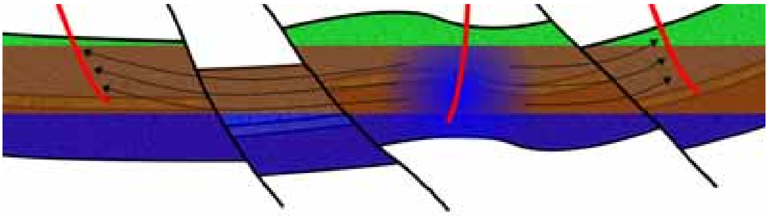


Figure 2.10: The same reservoir as in Figure 2.1, with water, oil and gas added as blue, brown and green colors respectively. There are two producer wells at the opposite side of the domain, while a single injector injects water to sweep the oil towards the producers. The wells are colored red.

The problem of simulating fluid flow and transport in porous media can generally be stated as a set of conservation equations for each component present in the porous medium. While this problem in principle can be formulated with an arbitrary number of phases and components, we will in this thesis only consider flow consisting of up to three phases, without any restrictions on the number of components present. Furthermore, we will assume that the *aqueous* phase is made up of a single component which is only present in the aqueous phase itself so that mixing behavior is restricted to the *liquid* and *vapor* phases. This formulation encompasses most commonly used models in reservoir simulation, including the standard black-oil model and many compositional models. By stating the conservation equations in the most general terms, we will see that several different model equations can be obtained.

Immiscible flow

To start, however, we consider a simplified equation where each phase contains a single component only, i.e., the distinction between phases and components does not exist. Let ρ_α and S_α be the density of phase α and the fraction of the void space occupied by phase α, respectively. First, let us assume that the phase velocity $\vec{v}_\alpha$ is known for all phases and that any source terms for each phase can be written as a function q_α. The conservation equation for the mass contained in a given phase can then be written for a given control volume Ω on standard form,

$$\int_\Omega \frac{\partial}{\partial t}\left(\phi \rho_\alpha S_\alpha\right) dV + \int_{\partial\Omega} \left(\rho_\alpha \vec{v}_\alpha\right) \cdot \vec{n}\, dS = \int_\Omega \rho_\alpha q_\alpha dV. \tag{2.1}$$

Letting the volume of the control volume tend towards zero and applying the divergence theorem to the surface integral we obtain the differential form of the equation,

$$\frac{\partial}{\partial t}\left(\phi \rho_\alpha S_\alpha\right) + \nabla \cdot \left(\rho_\alpha \vec{v}_\alpha\right) = \rho_w q_\alpha \tag{2.2}$$

valid for all points in the domain Ω.

Isothermal compositional flow

If we as the next step want to consider miscible flow, it is necessary to assume that there are two phases with N different components, which can exist in either phase, where X_i is the *mass fraction* of component i in the liquid phase and Y_i is the mass fraction of the same component in the vapor phase. If the mass of component i in phase α is m_i^α we can determine the mass fraction from,

$$X_i = m_i^\alpha / \sum_{j=1}^{N} m_j^\alpha. \tag{2.3}$$

Assuming that there is some known relationship for the mass transfer between phases, we obtain another differential equation for the conservation of the mass of a given component,

$$\frac{\partial}{\partial t}(\phi\left[\rho_l S_l X_i + \rho_v S_v Y_i\right]) + \nabla \cdot \left(\rho_l X_i \vec{v}_l + \rho_v Y_i \vec{v}_v\right) = \rho_l X_i q_l + \rho_v Y_i q_v \tag{2.4a}$$

$$\frac{\partial}{\partial t}\left(\phi \rho_w S_w\right) + \nabla \cdot \left(\rho_w \vec{v}_w\right) = q_w, \tag{2.4b}$$

where we have also included a conservation equation for the pure aqueous phase. These are the governing equations for isothermal compositional models, where the aqueous phase is immiscible.

Darcy's law for multiphase flow

It was sufficient to assume mass conservation to obtain (2.4), but not all terms are defined by mass conservation alone. The equations as stated require a velocity field for each phase. Flow in porous media is generally creeping flow without significant momentum and as such the fluid velocity will be modelled using a standard multiphase extension of Darcy's law,

$$\vec{v}_\alpha = -\lambda_\alpha \mathbf{K} \left(\nabla p_\alpha - \rho_\alpha g \nabla z \right), \qquad \lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}, \tag{2.5}$$

where $\mathbf{K}$ is some positive definite permeability tensor, p_α the phase pressure and $g\nabla z$ models the effect of gravity. Following convention, we have defined the ability of the phase to flow as the mobility λ_α, which is the ratio of the relative permeability $k_{r\alpha}$ and phase viscosity μ_α. The relative permeability is typically experimentally determined for a given mixture-medium pair. The viscosity can be obtained either experimentally or through correlations.

Relative permeability

The relative permeability $k_{r\alpha}$ for each phase models the ability of a certain fluid to flow through a specific porous media in the presence of other phases. The relative permeability is a monotone function of the phase saturation that takes on values between zero and unity, accounting for how easily each phase travels through the medium on a microscopic scale. Different fluids flowing through the same medium can have very different relative permeability curves, as the microscopic sweep of water, gas and liquid hydrocarbon can flow in different regions of the rock microstructure and stick to the pore walls to varying degrees.

There exist analytical expressions for relative permeability. In practice, the coefficients of these models are adjusted based on experiments performed on representative samples of the medium. In many cases, the relative permeabilities are created through the history-matching process, in which the numerical model of a reservoir is adjusted to reproduce real-world output. Examples of analytical expressions for relative permeabilities include the exponential Corey-type for a wetting and a non-wetting phase,

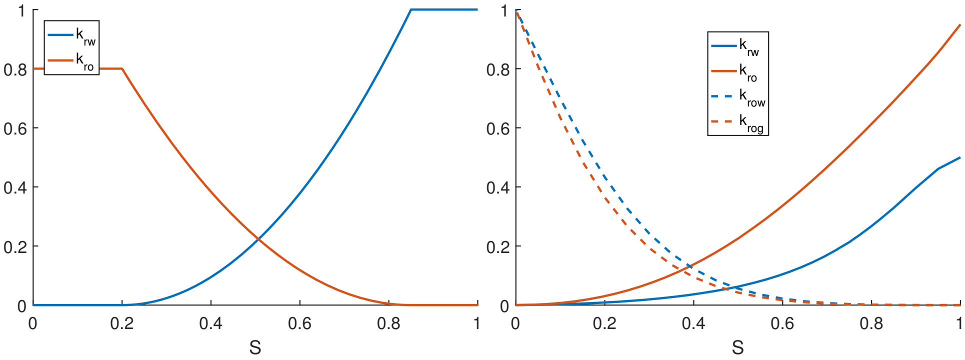
$$k_{rw}(S) = S_w^n, \quad k_{rn}(S) = (1 - S_w)^n, \tag{2.6}$$

as well as the form given by Brooks and Corey [14] which introduced a different exponent coefficient β,

$$k_{rw}(S) = S_w^{3+2/\beta}, \quad k_{rn}(S) = (1 - S_w)^2 \left(1 - S_w^{1+2/\beta}\right). \tag{2.7}$$

In this context, the wetting fluid is the phase that has the lowest contact angle between the medium surface and the droplet surface when a sample of the phase is placed on the medium.

Figure 2.11 demonstrates both a simple Corey-type relative permeability function, as well as a relative permeability system from a real field model. For problems with three phases present, $k_{r\alpha}$ is defined for each pair of phases, and the three-phase relative permeability model is evaluated by a combination of the different curves. Several choices for relative permeability in regions with true three-phase flow have been proposed [107, 106, 9].



(a) Analytical two-phase relative permeability

(b) Three-phase relative permeability from Norne

Figure 2.11: Two examples of relative permeability: A simple two-phase system based on Corey-exponents and the three-phase system from the Norne field model.

Capillary pressure

The observant reader may have noticed that even though we have defined most of the terms in the multiphase extension Darcy's law (2.5), we have still not considered the different phase pressures. The difference in phase pressures in porous media is normally the result of capillary action, whereby a combination of differences in surface tension and no-slip conditions for two different fluids will result in a potential difference. Capillary forces become more significant for smaller pore diameters and will over time introduce the spontaneous infiltration of the wetting fluid into the non-wetting region.

The form of capillary pressure used in MRST is given as a function of saturation separately for the oil-water and oil-gas contacts,

$$p_o - p_w = p_c^{ow}(S_w), \qquad p_g - p_o = p_c^{og}(S_g).$$

Note that there is by convention a difference in sign for the pressure change due to capillary action. One well-known example dataset is the SPE 9 model where the capillary pressure is significant and nearly discontinuous for the oil-water contact, as shown in Figure 2.12. The effect of capillarity is usually considered to be important to resolve at small scales and less significant at the field scale, where viscous and buoyancy forces dominate the flow regime.

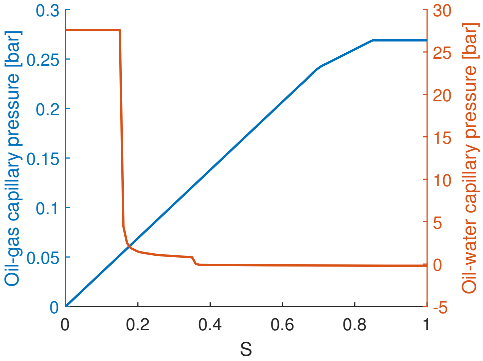


Figure 2.12: Capillary pressure example: The three-phase system for the SPE 9 model contains both capillary curves between the oil and water phases as well as the gas and oil phases, which differ in orders of magnitude.

2.4 Immiscible, incompressible flow

A commonly studied set of equations for flow and transport in porous media are the conservation equations for incompressible and immiscible flow. To derive these equations from (2.4) we will assume that the porosity is constant, that the densities are constants for each phase and that each component is only found in a single phase, i.e., that $X_i > 0$ only if $Y_i = 0$ and vice versa. By taking the sum over all components in phase and recalling that the mass fractions sum up to unity as per (2.3), we obtain conservation equations for each present phase,

$$\phi\frac{\partial}{\partial t}S_w + \nabla \cdot \vec{v}_w - q_w = 0, \tag{2.8a}$$

$$\phi\frac{\partial}{\partial t}S_l + \nabla \cdot \vec{v}_l - q_l = 0, \tag{2.8b}$$

$$\phi\frac{\partial}{\partial t}S_v + \nabla \cdot \vec{v}_v - q_v = 0. \tag{2.8c}$$

One popular solution strategy for these equations is the sequential pressure-transport scheme, which exploits the weak coupling between the spatially global pressure variable and the spatially local saturation variables. In the following, we will detail the pressure and transport equations individually, as they are the test problems used in several of the publications in this thesis. For details on the discretizations of these equations, see Chapter 3.

The pressure equation

The system (2.8) has a mixed elliptic-hyperbolic nature. To illustrate this, we can reformulate the system in terms of a pressure and a set of transport equations. The papers of this thesis exploit this reformulation to create efficient and robust numerical methods for coupled flow and transport. The pressure equation, under the simplified assumptions of incompressible flow and constant scalar permeability, is equivalent to the elliptic Poisson equation,

$$\nabla^2 p = q. \tag{2.9}$$

In general, however, this is not the case, and for more complex fluid physics, the pressure equation may be parabolic and will only be truly elliptic in the incompressible limit. In this section, we consider the incompressible pressure equation, which can be derived directly from the conservation equations for immiscible, incompressible flow (2.8). See Papers VIII and XIV for the black-oil and compositional pressure equations, respectively.

Regardless of the complexity of the model equations, the derivation principle remains the same: Ensure that the pressure equation is some weighted sum of the individual phase conservation equations, where the dependence on the time derivative of transported quantities is eliminated or minimized. To find the pressure equation from (2.8), we insert the velocity expressions from Darcy's law (2.5), assume a single unique pressure, neglect gravity and sum the equations together. We obtain the incompressible pressure equation,

$$0 = \sum_{\alpha}^{\{l,w,v\}} \left[\phi \frac{\partial}{\partial t} S_\alpha - \nabla \cdot (\mathbf{K} \lambda_\alpha \nabla p) - q_\alpha \right] \tag{2.10}$$

$$= \phi \frac{\partial}{\partial t} (S_l + S_v + S_w) - \nabla \cdot (\mathbf{K}(\lambda_l + \lambda_v + \lambda_w) \nabla p) - q_l - q_v - q_w \tag{2.11}$$

$$= \nabla \cdot (\mathbf{K} \lambda_t \nabla p) + q_t. \tag{2.12}$$

As noted earlier, the resulting equation is equivalent to the variable coefficient Poisson's equation when $\mathbf{K}$ is a scalar and is a model equation in many different fields, with varying physical interpretation of the variable coefficient. In our case, the variable coefficient consists of the scalar total mobility λ_t, which depends on fluid distribution in the medium, and the absolute permeability $\mathbf{K}$ which is a tensor representing the properties of the medium itself. This is an elliptic PDE, and as such the pressure field is global in nature, where changes in boundary conditions or source terms in one location will immediately change the pressure field in potentially all points of the domain.

We can also pose the same pressure equation in a slightly different form that emphasizes that the equations represents a steady-state balance of fluid flow,

$$\nabla \cdot \vec{v}_t = q_t, \qquad \vec{v}_t = -\mathbf{K} \lambda_t \nabla p. \tag{2.13}$$

This is referred to as the mixed form.

Fractional flow transport equation

The fractional flow transport equation for a given phase α can be written as,

$$\phi \frac{\partial}{\partial t} S_\alpha + \nabla \cdot (f_\alpha \vec{v}_t) - q_\alpha = 0, \tag{2.14}$$

where $\vec{v}_t$ is some total velocity field. This equation is of a hyperbolic nature, due to the finite speed of propagation for the saturation in the velocity field. In this form, the dependence on the transport upon the pressure field is through the total velocity. The fractional flow f_α for a phase is defined as the ratio of the mobility of the phase to the total mobility for all phases,

$$f_\alpha = \frac{\lambda_\alpha}{\lambda_w + \lambda_o + \lambda_g}. \tag{2.15}$$

Sequential solution of the pressure and transport

If we have solved (2.13) produces both the pressure field and the total velocity field $\vec{v}_t$. In addition to the pressure, we are usually interested in the fluid distribution for each phase as well, and one strategy is to then solve fractional flow transport equations (2.14), keeping the total velocity fixed, to advect the saturations along the velocity field. It is sufficient to solve the transport equations for two of the three transport equations, as the remaining phase saturation can be determined by the constraint $\sum_\alpha S_\alpha = 1$.

2.5 The black-oil equations

The workhorse of practical reservoir simulation is the black-oil equations. These equations represent an intermediate step between the full complexity of the compositional equations, in which each component is modeled individually and properties and phase behavior are predicted by equations of state, and immiscible, incompressible flow, where phases do not exchange mass with each other and properties are greatly simplified.

The standard formulation of the black-oil equations will allow the gas component to dissolve into the oil phase. The case of oil vaporizing into the gas phase, referred to as live-oil, is not considered here, but the effects are modeled in an analogous manner to gas dissolving in oil.

In the black-oil model, an auxiliary variable r_s is introduced. The definition of r_s is somewhat curious, as it for a given volume of the oil phase refers to the volume ratio between free gas and oil if that volume was transported to standard conditions. Standard conditions may refer to either pressure at the surface or at the gas separator of the field, depending on the dataset. The concept of standard conditions carries through to the definition of densities, where formation volume factors (FVF) are used to convert fluid volumes between reservoir and standard conditions,

$$\begin{aligned}
\rho_o(p, r_s) &= \frac{(r_s \rho_g^s + \rho_o^s)}{B_o(p, r_s)} = b_o(p, r_s)\left(r_s \rho_g^s + \rho_o^s\right), \\
\rho_g(p) &= \frac{\rho_g^s}{B_g(p)} = b_g(p)\rho_g^s, \\
\rho_w(p) &= \frac{\rho_w^s}{B_w(p)} = b_w(p)\rho_w^s.
\end{aligned}$$

Introducing the concept of surface volumes is especially useful when working with more advanced boundary conditions and source terms in the form of wells, where the injected and produced volumes are given at very different conditions than in the reservoir. Note that we here have used ρ^s_α to denote the density at standard conditions, which is a constant for each component. We have opted to use the reciprocal of the FVF ($b_\alpha = 1/B_\alpha$) in this description, which is sometimes incorrectly referred to as the *inverse formation volume factor*. We assume that there exists a tabulated maximum value for dissolved gas-in-oil as a function of pressure $r^{sat}_s(p)$ so that when the oil is fully saturated and $r_s \geq r^{sat}_s(p)$, the bubble-point pressure has been reached and free gas is formed. We can then write out one conservation equations for each phase, divide each equation by the corresponding density at standard conditions ρ^s_α and modify the gas component equation to account for the dissolved gas component to obtain the black-oil equations,

$$\frac{\partial}{\partial t}\left(\phi b_w S_w\right) + \nabla \cdot \left(b_w \vec{v}_w\right) = b_w q_w, \tag{2.16}$$

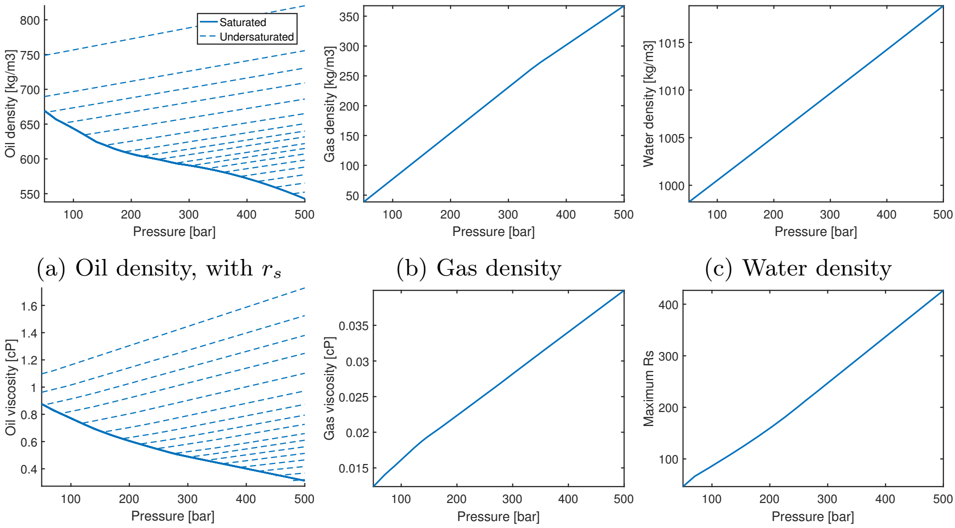
$$\frac{\partial}{\partial t}\left(\phi b_o S_o\right) + \nabla \cdot \left(b_o \vec{v}_o\right) = b_o q_o, \tag{2.17}$$

$$\frac{\partial}{\partial t}\left(\phi [b_g S_g + r_s b_o S_o]\right) + \nabla \cdot \left(b_g \vec{v}_g + R_s b_o \vec{v}_o\right) = b_g q_g + r_s b_o q_o. \tag{2.18}$$

Fluid properties in the black-oil model

The properties, hereunder formation-volume-factors and viscosities, are tabulated from either experimental data or from a compositional model with empirical correlations. For the oil phase, additional curves have to be specified for saturated and undersaturated flow, as the dissolved gas must result in density changes to be thermodynamically consistent. Coats [23] derived explicit inequalities for the b-factors and dissolved components to ensure positive partial volumes of each phase.

Figure 2.13 plots the PVT-properties from the SPE 1 [93] benchmark model. We note that while the gas and water phase properties are simple monotone functions of evaluation pressure, the oil phase is significantly more complex. The oil density appears to decrease with increasing pressure, which may at first seem counter-intuitive. This behavior is due to the oil phase being a mixture of heavy and light components and the whole line represents the *saturated* state, where there exists free gas that is gradually dissolved as the pressure increases. If the system runs out of gas to dissolve, the dashed *undersaturated* curves are followed and the density returns to the expected increase due to pressure. The swelling effect of gas dissolving into the oil phase as a function of pressure makes the accurate prediction of pressures more important for the black-oil model than for immiscible flow.



(a) Oil density, with r_s

(b) Gas density

(c) Water density

(d) Oil viscosity, with r_s

(e) Gas viscosity

(f) Maximum r_s

Figure 2.13: Example of black-oil type PVT-behavior taken from the SPE 1 benchmark dataset. Note that as the gas is allowed to dissolve into the oil phase, the oil properties contain functions for both the saturated case (where free gas exists and mixes instantaneously) and the undersaturated case (when the gas component only exists inside the oil phase). The water viscosity is a constant 0.31 cP for all pressure values and is consequently omitted from the plot.

2.6 Compositional properties

Compositional simulation is characterized by systems with N different components where phase behavior is predicted by an equation of state. The classical equation of state is the ideal gas law,

$$p = \frac{RT}{V_m}$$

where p is the pressure, V_m is the volume of one mole gas, T the temperature and R the ideal gas constant. For isothermal problems, the ideal gas law assumes a linear relationship between volume and pressure, which makes the density linear with increasing pressures. As this is not reasonable outside of specific pressure ranges for gases, there exists a large body of research on more accurate phase prediction for mixtures at different pressure and temperature ranges. One of the most common equations of state employed for hydrocarbons at relatively high pressures is due to Peng and Robinson

[96], which for a pure component can be written as

$$p = \frac{RT}{V_m - b} - \frac{a\alpha}{V_m(V_m + b) + b(V_m - b)}, \tag{2.19}$$

where the coefficients a, b, α are given by

$$a = \frac{0.45724R^2T_c^2}{p_c}, b = \frac{0.07780RT_c}{p_c},$$
$$\alpha = \left(1 + (0.37464 + 1.54226\omega - 0.26992\omega^2)(1 - T_r^{1/2})\right)^2.$$

The properties p_c and T_c refer to pressure and temperature taken at the critical point. For the reduced temperature, T_r, we define

$$T_r = \frac{T}{T_c}.$$

The critical point is where the notion of gas and liquid converge. ω is the acentric factor which is a non-negative constant indicating the how far the molecule is from spherical shape, where $\omega = 0$ indicates a completely spherical molecule. The degree to which an equation of state deviates from the ideal gas behavior of (2.6) can be measured by the *compressibility factor* Z,

$$p = Z\frac{RT}{V_m}. \tag{2.20}$$

Flash equations

We let X_i and Y_i be the mass fractions of component i in the liquid and vapor phases respectively, computed from the phase mole fractions for the liquid (x_i) and vapor (y_i) phases,

$$X_i = \frac{x_iW_i}{\sum_{j=1}^N x_jW_j}, \qquad Y_i = \frac{y_iW_i}{\sum_{j=1}^N y_jW_j}, \tag{2.21}$$

where we have assumed that the molar weights for each component W_i is known. Based on the equation-of-state, we must find the distribution of components in each phase as well as the liquid mole fraction L from the overall composition $\mathbf{z} = [z_1, ..., z_N]^T$. These quantities can be obtained by solving the *flash equations*, here given in the same form as in [118],

$$\begin{aligned} f_{iv}(p, T, x_1, ..., x_n, Z_l) - f_{il}(p, T, y_1, ..., y_n, Z_v) = 0, \text{ for } i \in \{1, ..., N\} \\ z_i - Lx_i - (1 - L)y_i = 0, \text{ for } i \in \{1, ..., N\} \\ \sum_{i=1}^N x_i - y_i = 0. \end{aligned}$$

The flash equations require that the component fugacities f_{il}, f_{iv} for both phases present are equal as well as ensuring that the moles of a component in each phase balance to the total amount of component moles. The need for flash can be determined a priori using a phase stability test to determine if a given composition under current conditions will split into two phases [80, 81]. Since the equations consist of $2N + 1$ equations and we need to obtain N component fractions for both phases as well as the liquid ratio, we can linearize and solve this system using either Newton's method or the method of successive substitution, in which equilibrium constants are successively updated based on the fugacity ratio. The fugacities and compressibility factor are predicted using a generalized cubic equation of state reformulated in terms of the compressibility factors Z_α (2.20),

$$Z_\alpha^3 + aZ_\alpha^2 + bZ_\alpha + c = 0, \tag{2.22}$$

where a, b, c are coefficients that depend on the phase composition and the type of equation of state used. This specific generalized cubic form was derived from Martin's equation [78] by Coats [22]. This polynomial form is convenient as the coefficients can easily be chosen to reduce to several well known cubic equations of state, e.g., Zudkevitch and Joffe [130], Redlich and Kwong [98], Redlich-Kwong-Soave [105], or Peng and Robinson [96]. Different equations may have mixing coefficients that are empirical and specific to that equation, so the common implementation is beneficial when working with existing test problems that cannot easily be used with any equation of state.

Saturations and phase properties

Once the flash equations have been solved, the phase saturations S_l and S_v can be computed by letting the compositional phases fill up the remaining pore volumes not saturated with the aqueous phase,

$$S_l = (1 - S_w)\frac{LZ_l}{VZ_v + LZ_l}, \qquad S_v = (1 - S_w)\frac{VZ_v}{VZ_v + LZ_l}, \quad V = (1 - L). \tag{2.23}$$

The compressibility factors represent the deviation of a phase's volumetric expansion behavior predicted by the ideal gas law. From the definition of density we can obtain explicit expressions for the liquid and vapor as

$$\rho_l = \frac{p}{RTZ_l}\sum_{i=1}^{N} x_i W_i, \qquad \rho_v = \frac{p}{RTZ_v}\sum_{i=1}^{N} y_i W_i. \tag{2.24}$$

Table 2.1: The four component mixture dataset used to illustrate compositional phase properties. We have not included empirical interaction coefficients between the components and consequently this table contains all required values for a compositional phase prediction using a cubic equation of state.

Component	T_c [K]	p_c [Bar]	V_c [m^3/mol]	ω	Mass [kg^3/mol]
Methane	190.56	46.00	$9.86 \cdot 10^{-5}$	0.011	0.016
Nitrogen	126.19	33.96	$8.94 \cdot 10^{-5}$	0.037	0.028
n-Pentane	469.70	33.70	$3.11 \cdot 10^{-4}$	0.251	0.072
n-Decane	617.70	21.03	$6.10 \cdot 10^{-4}$	0.488	0.142

To illustrate the emergent phase behavior from an equation of state, we will consider a simple four component mixture shown in Table 2.1. The open source CoolProp package was used to look up these properties [10]. The mixture is assumed to be comprised of equal molar fractions Methane, Nitrogen and the normal isomers of Pentane and Decane. From the table, we can see that the properties of the components vary significantly, where the heavier components generally have higher critical values, as well as acentric factors due to their molecules being longer chains of hydrocarbons. We compute the liquid saturation for this mixture, shown in Figure 2.14 using the Peng-Robinson equation of state in MRST. For low pressures, the mixture forms a vapor phase. As the pressure rises, the gas phase is gradually converted into liquid. At the critical point, the distinction between the vapor and liquid phases disappear and beyond this point the saturation is discontinuous. The complete code, requiring only the names of the individual components as input, can be written in 10 lines of code:

```
mrstModule add compositional
components = {'Methane', 'Nitrogen', 'n-Pentane', 'n-Decane'};
fluid = CoolPropsCompositionalFluid(components);
% Intialize  cubic eos  model
EOS = EquationOfStateModel([], fluid);
% Use standalone routine to get phase diagram for pressure and temperature ranges
z = [0.25, 0.25, 0.25, 0.25];
p = (0:1:250)*barsa;
t = 273:650;
[L, x, y, Z_L, Z_V, P, T] = getFlashTable(p, t, z, EOS);
% Liquid saturation
S_L = Z_L.*L./(Z_L.*L + (1-L).*Z_V);
% Plot the saturation with 25 contour lines
contourf(T, P/barsa, S_L, 25)
```

Once we have obtained the compressibility factors Z_L, Z_V and the liquid mole fraction L we obtain the densities using (2.24) and the viscosities are

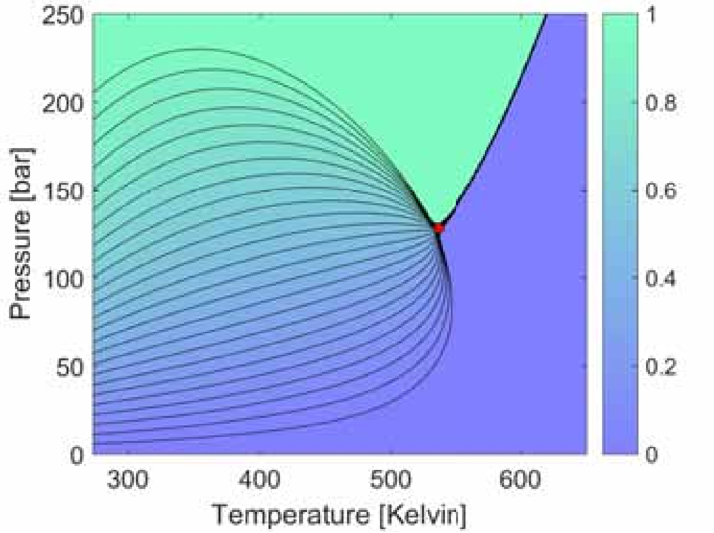
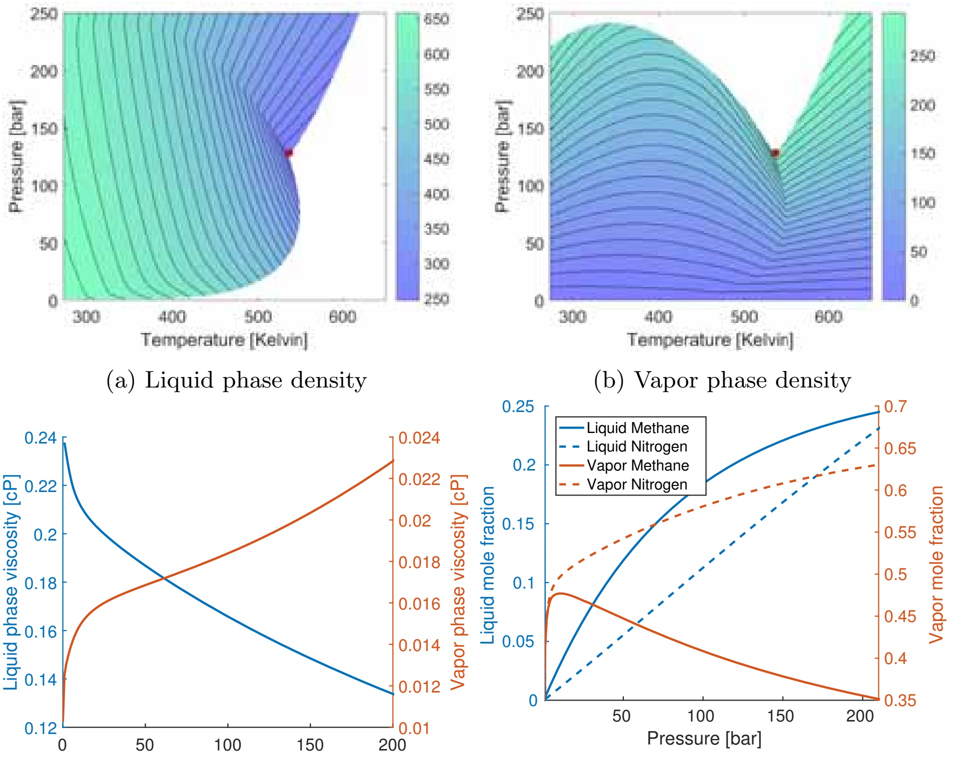


Figure 2.14: Predicted liquid saturation S_l for the four component test mixture. The critical point where the distinction between liquid and vapor phases disappear is marked with a red dot.

found using the Lohrenz-Bray-Clark correlation [70]. This is done by calls to the member functions of the `EquationOfState` class instance,

```
% Compute density
rhoL = EOS.computeDensity(P, x, Z_L, T, true); % Liquid
rhoV = EOS.computeDensity(P, y, Z_V, T, false); % Vapor
% Get viscosities
muL = EOS.computeViscosity(P, rhoL, T, x, true); % Liquid
muV = EOS.computeViscosity(P, rhoV, T, y, false); % Vapor
```

We have plotted the density and viscosities in Figure 2.15. For the phase densities, it is worth noting that for the lower pressures the difference between the two phases is large. As the pressure and temperature approaches the critical point, the liquid and vapor densities converge to the same values. The two lighter components Methane and Nitrogen are shown in Figure 2.15d for a fixed temperature of 350 K. As the pressure increases, the molar fraction of methane in the vapor phase gradually decreases as other components are vaporized, while the heavier nitrogen increases its share, which is also reflected in the evaluated viscosity shown in Figure 2.15c.



(a) Liquid phase density

(b) Vapor phase density

(c) Viscosities for $T = 350$ K

(d) Light component fractions for $T = 350$ K

Figure 2.15: The liquid 2.15a and vapor 2.15b density diagrams as a function of pressure and temperature for constant overall composition. As the pressure and temperature goes towards the supercritical region, the densities converge to the same value. The viscosity of the mixtures 2.15c also change as the components are transferred between the phases 2.15d.

Chapter 3

Spatial and temporal discretization

Analytical solutions of the different model equations introduced in Chapter 2 is not always feasible for general domains. For this reason, we will need to discretize the equations and solve the resulting nonlinear systems *numerically.* In this chapter, we will briefly consider some of the standard techniques used to solve flow and transport numerically that will be used in Part II of the thesis.

3.1 Spatial discretization

We have in the preceding chapter introduced the mixed elliptic-hyperbolic model equations and the grid. Once a domain has been divided into grid cells Ω_i as seen in Figure 3.1, a discretization can be applied to obtain discrete equivalents of the model equations. We will first consider the elliptic model problem before considering the hyperbolic and mixed-hyperbolic equations in a later section. The classical choices from numerical analysis for elliptic equations are broadly speaking different types of finite-element, finite-difference and finite-volume methods. For flow problems on models of geological scales, finite-volume schemes are popular due to exact local mass conservation and ease of implementation for cells with general polyhedral shapes. Assuming that we have some continuously differentiable velocity field $\vec{v}$, equations on the form (2.13) can be transformed into a surface integral by the divergence theorem,

$$\int\limits_{\Omega_i} (\nabla \cdot \vec{v})\, dV = \int\limits_{\partial\Omega_i} (\vec{v} \cdot \vec{n})\, dA = \int\limits_{\Omega_i} q\, dV. \tag{3.1}$$

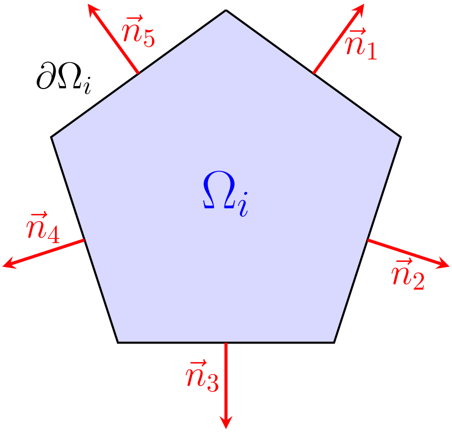


Figure 3.1: A single control-volume Ω_i, with boundary $\partial\Omega_i$ and normal vectors for each interface $\vec{n}_j$.

Two-point flux approximation

We want to integrate the fluxes over the control-volumes representing individual cells, and we thus require an expression for the flux at the interface between two cells. There are many possible flux expressions that lead to different schemes. A through comparison of the options available is outside the scope of this thesis, but we will consider the most commonly used approximation in some detail and compare it to another similar scheme numerically. The industry standard discretization is one of the simplest possible, where the flux over the interface is approximated using a two-point stencil, referred to as the two-point flux approximation (TPFA). Let the the interface between cells Ω_i and Ω_j in Figure 3.2 be denoted $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, and let x_f be the face centroid, A_f the area of the face and $\vec{n}_f$ be the unit normal vector of the interface. Then the flux through the interface can be approximated by,

$$\int\limits_{\Gamma_{ij}} \vec{v} \cdot \vec{n}\, dA \approx A_f \vec{v}(x_f) \cdot \vec{n}. \tag{3.2}$$

The flux at the interface can be approximated by a two-point difference approximation using Darcy's law for the pressure at the face centroid $p(x_f)$ and the cell centroid pressure in the left cell $p(x_i)$,

$$\vec{v}(\vec{x}_f) = -\mathbf{K}_i \nabla p \approx -\mathbf{K}_i \frac{\vec{c}_i}{|\vec{c}_i|^2} \left(p(x_f) - p(x_i)\right) = -T_{if} \left(p(x_f) - p(x_i)\right), \tag{3.3}$$

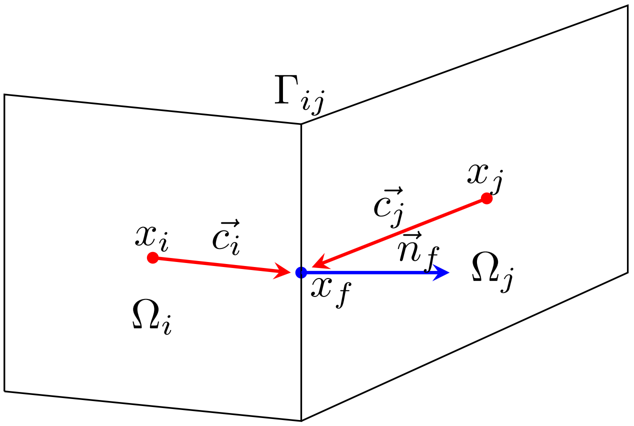


Figure 3.2: Two control-volumes Ω_i and Ω_j, along with their centroids x_i and x_j. The face centroid x_f is used to define the cell-face centroid vectors $\vec{c_i}$ and $\vec{c_j}$ that are required to derive the two-point flux approximation.

where we have defined $\vec{c_i}$ as the vector from cell-centroid to face-centroid, as well as the half-face transmissibility T_{if}. There exists an analogous expression for the same velocity seen from the other side of the interface where the only difference is a change in sign,

$$\vec{v}(\vec{x}_f) \approx \mathbf{K}_j \frac{\vec{c_j}}{|\vec{c_j}|^2} \left(p(x_f) - p(x_j)\right) = T_{jf} \left(p(x_f) - p(x_j)\right). \tag{3.4}$$

If we assume that the pressure is continuous at the interface, we can rearrange the two equations to eliminate the face pressure $p(x_f)$,

$$\vec{v}(x_f) = \frac{1}{T_{jf}^{-1} + T_{if}^{-1}} (p(x_i) - p(x_j)), \tag{3.5}$$

which can then be combined with (3.2) to obtain the final expression for the two-point flux,

$$\int\limits_{\Gamma} \vec{v} \cdot \vec{n}\, dA \approx \frac{A_f}{T_{jf}^{-1} + T_{if}^{-1}} \cdot \vec{n} \left(p(x_i) - p(x_j)\right) = T_{ij}(p(x_i) - p(x_j)). \tag{3.6}$$

The resulting finite-volume scheme for the pressure equation (2.13) is then a discrete analogue of (3.1),

$$\sum_{j \in N_i} V_{ij} = \sum_{j \in N_i} T_{ij} \left(p(x_i) - p(x_j)\right) = Q_i,$$

where V_{ij} is the discrete flux between cells i and j and N_i is the set of all cell neighbors for cell i. Q_i is the volume-integrated source terms present in the cell.

Each cell will in the TPFA scheme only be connected to the neighbors with which it shares faces. For the pressure equation discretized on quadrilaterals in 2D with a homogeneous isotropic medium, this scheme reduces to the classical five-point stencil used to solve Poisson's equation. The approximation in (3.3) neglects the component of the flux not tangential to the cell-face centroid vector $\vec{c}_i$ and the scheme is only consistent for K-orthogonal grids, i.e., grids where the flow direction aligns with the cell-face centroid vectors,

$$(\mathsf{K}\vec{n_f}) \times \vec{c}_i = 0 \quad \forall\, i. \tag{3.7}$$

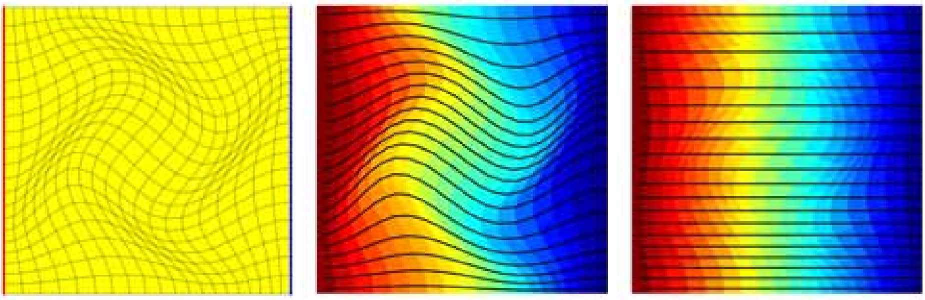
Consistent discretizations for the flux for general grids is an active field of research. Many different schemes have been suggested, including different multipoint flux approximation (MPFA) schemes [3, 4, 27], mixed finite-element methods [57], mimetic finite-difference methods [13, 69] and more recently nonlinear TPFA-schemes for which the discretization itself depends on the unknown pressure field [60, 88].

We can demonstrate this with a simple example with a grid that violates the assumptions of K-orthogonality. We first define a 20×20 fine grid on the unit square and apply a boundary condition $p = 1|_{x=0}$ and $p = 0|_{x=0}$ with implicitly defined no-flow boundary conditions $(\vec{v} \cdot \vec{n} = 0|_{y=1 \vee y=0})$ for the remaining boundaries. The grid coordinates are then distorted using the `twister` routine.

```
% Define grid and set boundary conditions
G = cartGrid([20, 20], [1, 1]);
G = computeGeometry(twister(G, .1));
bc = pside([], G, 'xmin', 1);
bc = pside(bc, G, 'xmax', 0);
rock = makeRock(G, 1, 1);
```

We can then solve the same problem using both the MPFA-O and the TPFA solvers. Note that since the permeability is a constant scalar, the pressure equation reduces to Poisson's equation, which has an analytical solution of a linear drop from the left to right for this problem, $p(x) = 1 - x$.

```
T_tpfa = computeTrans(G, rock); % TPFA transmissibility
T_mpfa = computeMultiPointTrans(G, rock); % MPFA transmissibility
state = initResSol(G, 0); % Define state
fluid = initSingleFluid('rho', 1, 'mu', 1); % Trivial single-phase fluid
tpfa = incompTPFA(state, G, T_tpfa, fluid, 'bc', bc); % Solve TPFA
mpfa = incompMPFA(state, G, T_mpfa, fluid, 'bc', bc); % Solve MPFA
```



(a) Twisted grid (b) TPFA (c) MPFA

Figure 3.3: Not all discretizations are equal. A twisted grid with linear boundary conditions (left) leads to consistency issues for the two-point flux approximation method (middle), while the consistent multipoint flux-approximation (right) does not suffer from grid orientation effects and agrees with the analytical solution of a linear pressure drop from $x = 0$ to $x = 1$. Note that the pressure is plotted as constant for each grid cell, which makes it not appear linear between cell centroids.

The results, as well as the twisted grid, are shown in Figure 3.3. We observe that the pressure field for the TPFA solution is distorted where the grid cells deviate from regular squares. The streamlines, tangential to the flux field, are shown in black, and demonstrate that the MPFA scheme accurately resolves the flow field for this problem. It should be noted, however, that the MPFA-method can have monotonicity issues for certain grids [89].

Upwinding of advected quantities

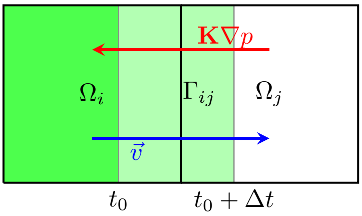


Figure 3.4: Conceptual illustration for the first-order upwind scheme. The fluid, shown in green, sweeps through the domain as the time progresses. To evaluate functions that depend strongly on the fluid saturation on the interface, a stable choice is to approximate the saturation by the inflow cell Ω_i.

We have obtained an expression, (3.6), which relates the pressure drop to the fluxes. Our objective is to find a discrete version of (2.5). If we insert the discrete flux expression into (3.6) integrated over a face, we obtain

$$(V_{ij})_\alpha = -(\lambda_\alpha)_{ij} T_{ij} \left(p(x_i) - p(x_j) - (\rho_\alpha g \nabla z)_{ij} \right). \tag{3.8}$$

The buoyancy force is straightforward to evaluate discretely, as the phase density is constant and we can obtain the gradient of the depths of the cell centroids at the interface,

$$(\nabla z)_{ij} \approx \frac{z(x_j) - z(x_i)}{|x_j - x_i|}. \tag{3.9}$$

The mobility λ_α must be carefully evaluated, however, as it strongly depends on the saturation, which is a transported quantity. Consider a two-cell model as shown in Figure 3.4. As the front moves from the left cell and into the right, the saturation at the interface will change. To avoid over and undershoots in the saturation, one natural choice is to upwind the value at the interface, that is, take the value of the mobility to be equal to the cell from which the front is propagating,

$$(\lambda_\alpha)_{ij} \approx \begin{cases} (\lambda_\alpha)_i, & \text{if } (V_{ij})_\alpha > 0, \\ (\lambda_\alpha)_j, & \text{if } (V_{ij})_\alpha \leq 0. \end{cases} \tag{3.10}$$

This is the standard first-order upwind scheme. Much like the TPFA scheme, it is straightforward to implement for general grids and does not introduce oscillations. Higher-order schemes can be used for applications where more accurate saturation profiles are required.

Discretization of wells

In the preceding sections, we have always assumed that the flow is driven by some volumetric source term q. Flow in reservoir simulation can be driven by Dirichlet or Neumann boundary conditions, but more often than not, the displacement of fluids is controlled using wells as source terms. Wells are of particular interest for subsurface applications, since reservoirs are normally closed off far below the caprock and wells represent the only option for injection or extraction of fluid, whether the application is oil recovery, gas storage, CO_2 sequestration or geothermal energy.

A well is a cylindrical hole with a diameter naturally measured in centimeters or inches through the medium that is connected to control facilities topside. The well-bore is very small compared to the typical grid blocks used in a reservoir model, with near-well flow happening at a completely different length scale than the rest of the model, and consequently numerical modelling can be a challenging prospect. The well may perforate multiple blocks in the simulation grid, where each perforated cell will potentially contain source terms. The connection between the well-bore and the reservoir is defined by an expression very similar to (3.6) for a given phase α,

$$q_i^\alpha = \begin{cases} W_I \gamma_\alpha \lambda_{ti}(p_i - p_{bh} + \rho_{mix} g \Delta z), & \text{if } q_i > 0, \\ W_I \lambda_{\alpha i}(p_i - p_{bh} + \rho_{mix} g \Delta z), & \text{if } q_i < 0, \end{cases} \tag{3.11}$$

where we differentiate between completions that inject and perforations that produce the phase in question. Injector cells are weighted by the total mobility in the cell λ_{ti} and a factor γ_α which corresponds to the volumetric fraction of that fluid in the injector. For instance, if a well injects pure water $\gamma_i = \delta_{wi}$. For the producer cells, the well will produce the fluids present in the grid cell according to the mobilities of the different phases. The pressure inside the well is normally defined at a single point referred to as the bottom-hole pressure. The pressure in the well in each cell along the wellbore is computed using standard hydrostatic conditions, with a mixture ρ_{mix} density that reflects the average composition of the well above the current perforation. The standard wells in MRST use the assumption of instantaneous mixing for well flow, which means that the well bore composition immediately changes when the inflow or outflow conditions change. For the well indices W_I, which are analogous to interblock transmissibilities, we follow the seminal work of Peaceman [95] who developed analytical expressions for hexahedral grid blocks.

Figure 3.5 demonstrates a conceptual drawing of a single injector well. Once the well has been discretized, we need to define a control equation for each well that allows some combination of the phase flow rates, the total flow rate and the bottom-hole pressures to match the observed controls at the surface. If a well is controlled by the bottom hole pressure, the control equation is trivial,

$$e_c = p_{bh}^t - p_{bh} = 0, \tag{3.12}$$

where p_{bh}^t is some target bottom-hole pressure. For volumetric rates, the control must account for the total flow for all perforations,

$$e_c = q^{t\alpha} - \sum_{i=1}^{n} q_i^\alpha = 0, \tag{3.13}$$

where α can be either phase, or some combination of phases. For instance, liquid rate controls targeting the sum of produced water and liquid hydrocarbons is a typical control employed by producers with finite liquid capacity topside.

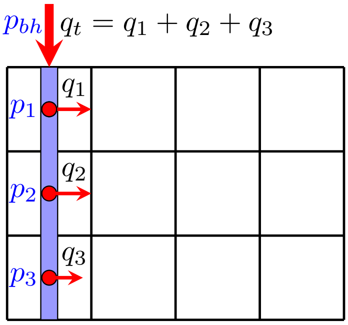


Figure 3.5: Conceptual model of a well. The well perforates three different cells, and the well model assigns a single flux to each perforation.

3.2 Temporal discretization

Newton's method for nonlinear equations

Although certain problems from flow and transport in porous media have limit solutions that are linear, e.g., the incompressible pressure equation(2.13) or the transport equations (3.16) with linear flux functions, these seldom appear in practice (or only under very specific conditions). The problems we are interested in solving numerically are therefore predominately systems of nonlinear equations.

Let $\mathbf{R}(\mathbf{x})$ be a vector valued function of some primary variable $\mathbf{x}$ where we want the solution $\mathbf{R}(\mathbf{x}) = 0$. Newton's method is the natural technique for the solution of this system. If we now define the Jacobian $\mathbf{J}(\mathbf{x})$ as the matrix $d\mathbf{R}(\mathbf{x})/d\mathbf{x}$, we can apply Newton's method for the linearized update to the primary variables,

$$-\mathbf{J}(\mathbf{x}_i)\Delta\mathbf{x}_i = \mathbf{R}(\mathbf{x}_i), \qquad \Delta\mathbf{x}_i = (\mathbf{x}_{i+1} - \mathbf{x}_i)\,. \tag{3.14}$$

For problems where Newton's method converges unconditionally, it is straightforward to iteratively update $\mathbf{x}$ until $\|\mathbf{R}(\mathbf{x})\| < \epsilon$ for some chosen norm $\|-\|$ and tolerance ϵ. Perhaps due to the quadratic convergence rates under certain conditions, Newton's method forms the backbone for the solution of nonlinear equations in many fields, and reservoir simulation is no exception. One of the major challenges is then how to best choose the systems of equations to solve in order to exploit strong and weak couplings, and isolate subproblems that can be efficiently solved by appropriate linear solvers.

Three different strategies for three-phase flow

We will consider the conservation equations (2.8) introduced in Chapter 2 for three-phase immiscible transport where all phases are incompressible,

$$\begin{aligned} \phi\frac{\partial}{\partial t}S_w + \nabla \cdot \vec{v}_w - q_w &= 0 \\ \phi\frac{\partial}{\partial t}S_l + \nabla \cdot \vec{v}_l - q_l &= 0 \\ \phi\frac{\partial}{\partial t}S_v + \nabla \cdot \vec{v}_v - q_v &= 0, \\ S_w + S_l + S_v &= 1 \end{aligned}$$

These equations are somewhat simplified compared to the compositional or black-oil cases, but in terms of solution strategy, our options are very similar. For four equations, we want to solve for four primary variables. For this simplified model, the obvious choices are fluid pressure p and phase saturations S_w, S_l, S_v. If we use the closure relation for saturations, we can further reduce the system to three equations and three variables, p, S_w, S_v.

Fully implicit

The most obvious approach is to simply linearize (2.8) directly for our choice of primary variables and use backward Euler to discretize the accumulation terms to obtain the fully-implicit system on semidiscrete form,

$$\begin{aligned} \frac{\Phi}{\Delta t}(S_w^{n+1} - S_w^n) + \nabla \cdot \vec{v}_w(p^{n+1}, S_w^{n+1}) - q_l &= 0 \\ \frac{\Phi}{\Delta t}(-S_v^{n+1} - S_v^{n+1} + S_v^n + S_v^n) + \nabla \cdot \vec{v}_l(p^{n+1}, S_w^{n+1}, S_v^{n+1}) - q_l &= 0 \\ \frac{\Phi}{\Delta t}(S_v^{n+1} - S_v^n) + \nabla \cdot \vec{v}_v(p^{n+1}, S_v^{n+1}) - q_v &= 0, \end{aligned}$$

where the $n+1$ superscript indicates that the quantities are evaluated at the end of the time-step[1]. The fully-implicit scheme is unconditionally stable for decaying problems and has been applied to a wide range of reservoir simulation problems. The scheme is also computationally expensive as the resulting linear system is $3M \times 3M$ in size, where M is the number of cells. The conservation equation for each phase contains both a coupled

[1] For brevity, we have assumed that the source terms themselves do not depend on the primary variables, which is seldom true in practice. Their treatment for well models are in practice analogous to the velocity/flux-expressions.

pressure problem (due to Darcy's law for each cell-cell interface, and every phase) and a transport subsystem (due to upwinding of the saturations in the velocity field). The sparsity pattern of the Jacobian matrix for a small 10 cell problem is shown in Figure 3.6, where we can see that the pressure subsystem bears resemblance to a standard second-order discretization of the Laplacian, while the saturation derivatives depend solely on the flow direction.

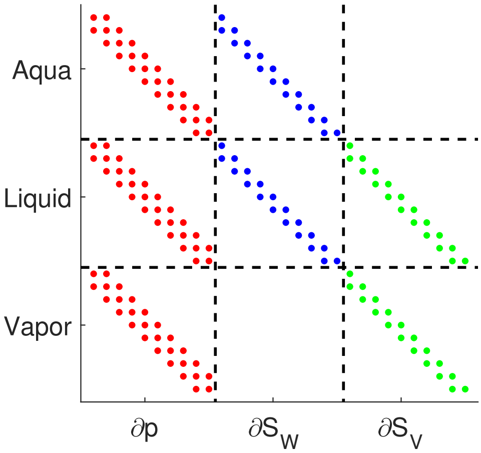


Figure 3.6: Sparsity pattern of a three-phase problem with 10 cells discretized with a fully-implicit scheme. All three equations have a strong dependence on the pressure, while the saturation derivatives are not coupled to the same degree, at least in terms of sparsity pattern.

Implicit pressure

For models with a large number of grid cells, the fully-implicit method may no longer be computationally feasible due to the size of the resulting linear systems that need to be inverted. The coupling between the saturations and the pressure variable is weak for many problems, and consequently one option is to first solve the pressure equation (2.13),

$$\nabla \cdot (\mathbf{K}\lambda_t \nabla p) + q_t = 0.$$

If we approximate the saturation dependency by the values from the previous time-step $S_l^{n+1} \approx S_l^n, S_v^{n+1} \approx S_v^n$, and discretize the equation implicitly with respect to the pressure variable we obtain,

$$\nabla \cdot \left(\mathbf{K}\lambda_t(S_l^n, S_v^n)\nabla p^{n+1}\right) + q_t^{n+1} = 0,$$

which is equivalent to the variable coefficient Poisson's equation with a tensor, known to be a linear equation. This equation has the same sparsity structure as either of the three pressure blocks in the Jacobian of the fully-implicit nonlinear system, which is significantly more expensive to solve.

Explicit saturation

Once we have solved for a new pressure and a velocity field, we need to update the phase saturations. The first option is to treat the saturations explicitly for the fluxes in the transport, making the saturation update for each phase trivial with a forward Euler scheme,

$$S_\alpha^{n+1} = S_\alpha^n + \frac{\Delta t}{\Phi}\left[\nabla \cdot \vec{v}_\alpha(p^{n+1} S_w^n) - q_\alpha^{n+1}\right].$$

Treating the variables explicitly leads to conditional stability by the Courant-Friedrichs-Lewy (CFL) condition for each cell,

$$\frac{\Delta t}{\Phi_i} F_i \leq 1, \tag{3.15}$$

where F represents the magnitude of the flow terms between the cells [20]. If the CFL condition is violated, displacement fronts (or material waves) may pass completely through the cell within a single time step so that updated saturation values depend on more cells than their next neighbors, and the approximation $\vec{v}_\alpha(p^{n+1} S_w^{n+1}) \approx \vec{v}_\alpha(p^{n+1} S_w^n)$ is no longer valid. Time-steps must be chosen carefully to avoid stability issues [21]. As the denominator contains the pore volume of the cells, very small cells or cells with very low porosity may lead to strict time-step restrictions for some models.

Implicit saturation

Instead of using an explicit scheme for the phase saturations, it is possible to use an implicit scheme. For the backward Euler time-discretization with fixed pressure and total velocity, we obtain the classical fractional flow (2.14) formulation of the transport equations on semi-discrete form,

$$\frac{\Phi}{\Delta t}\left(S_\alpha^{n+1} - S_\alpha^n\right) + \nabla \cdot \left(f_\alpha^{n+1} \vec{v}_t^{n+1}\right) - q_\alpha = 0, \tag{3.16}$$

where the fractional flow-term is updated as the saturations change in the transport equations,

$$f_\alpha^{n+1} = \frac{\lambda_\alpha(S_\alpha^{n+1})}{\lambda_w(S_w^{n+1}) + \lambda_l(S_l^{n+1}) + \lambda_v(S_v^{n+1})}. \tag{3.17}$$

while the total velocity is kept fixed from the pressure equation,

$$\vec{v}_t^{n+1} = \vec{v}_w(p^{n+1}, S_w^n) + \vec{v}_l(p^{n+1}, S_l^n) + \vec{v}_v(p^{n+1}, S_v^n). \tag{3.18}$$

The consequence is that the total volumetric exchange between cells is kept fixed throughout the transport step, but the fraction of each phase changes according to the fractional flow term.

For the black-oil and compositional equations, Trangenstein and Bell [111, 112] proved that the sequential form of the equations without capillary forces under certain assumptions admits a parabolic pressure equation with hyperbolic transport equations. The treatment of upstream directions for the transport equations with fixed total velocity was detailed by Brenier and Jaffré [12]. The fine details of the schemes may differ, but many different authors have considered treating pressure and transport sequentially [28, 123, 111]. For problems where the total velocity depends strongly on the saturation variables, it is possible to re-visit the pressure equation after saturations have been updated.

Some authors also consider the adaptive implicit method (AIM) where cells estimated to be below the CFL limit are treated explicitly [110, 29, 99, 24, 128] as a middle ground between explicit and implicit methods when high flow rates is localized to specific regions near source terms and/or with high permeability.

Chapter 4

Multiscale solvers

4.1 Exact and inexact solvers

A discretized problem, whether it is linear or nonlinear, often amounts to solving one or more linear systems. The size and the difficulty of the linear systems depend on the discretization schemes used as well as the degree to which different types of physics is included. For instance, the fully-implicit scheme with a two-point flux approximation for a three-phase problem with a million cells will result in linearized systems with three million unknowns. During each time-step, solving a number of these systems may be required to obtain convergence, and a simulation of relevant time-scales may consist of hundreds of individual steps. As a result, reducing the time spent solving linear systems is one effective way of speeding up reservoir simulation problems.

In one manner of categorizations, solvers for flow and transport in porous media can be divided into exact and inexact solvers. Exact solvers solve the linear systems to a strict tolerance and represent classical direct and iterative solvers. Due to the size and memory requirements of the linear systems, iterative solvers are used in most modern simulators. Eclipse [101], a popular commercial reservoir simulator uses the orthomin iterative solver [117] with nested factorization as a preconditioner [6]. The constrained pressure-residual (CPR) preconditioner [120, 17] for the fully-implicit system is a popular, modern choice for reservoir simulation used in both research [51, 59] and commercial simulators [26, 16, 17]. CPR exploits the structure of the equations to decouple and solve an pressure equation from the linear system, which can then efficiently be solved by for example algebraic multigrid [108]. The pressure solution is then used to precondition the remainder of the system which has weaker global coupling between variables. How to efficiently decouple the pressure equation algebraically, and what solver stages to use, is an area of active research [32, 15].

In the other end of the spectrum we find inexact solvers. Inexact solvers are formulated from the assertion that the full problem may be too expensive to solve completely, and consequently one should be satisfied with a non-converged solution that captures the most important features of the fine-scale solution. This is not to say that regular linear solvers are always set to converge to machine precision, but rather that there is a philosophical difference between the two classes. Multiscale solvers are fundamentally approximate solvers, even if they for some configurations *can* converge to machine precision if required.

4.2 Multiscale solvers

Multiscale methods for elliptic problems with varying coefficients are introduced in the form of the multiscale finite element method MsFEM [39]. A discretization method with subgrid coefficient resolution is attractive for flow in porous media, where large variation in medium properties is the norm. Several articles on multiscale methods therefore use porous media as their model problem. In this thesis, the starting point is the multiscale finite-volume (MsFV) method [44]. We will in this section give a brief overview of the multiscale concept by using the MsFV-framework as an example. For a more detailed review of multiscale solvers, we defer the reader to the summary of individual papers in Chapter 6, as well as the introduction of Paper XII.

Multiscale solvers for flow

Assume that we have a fine-scale problem stemming from the discretization of the incompressible pressure equation (2.13) by a cell-centered finite-volume scheme. The fine-scale problem takes the form of a set of linear equations for the pressure in each cell,

$$\mathbf{A}\mathbf{p} = \mathbf{q}. \tag{4.1}$$

This system can be challenging to solve directly because it represents an elliptic or near-elliptic equation, where small changes to the solution vector can have a large impact on the norm of the residual. Due to several orders of magnitude variation in the matrix coefficients, such systems can be ill-conditioned. For large models, it will contain millions of degrees of freedom that are highly coupled, as elliptic equations have infinite speed of propagation. Any changes in one part of the domain will potentially immediately affect all other points in the domain to some degree. In order to reduce the computational cost of solving (4.1), we will instead only solve it *approximately*, while still retaining certain key physical properties of the solution. Namely, we will try to find some approximate pressure $\mathbf{p}_f \approx \mathbf{p}$ that is closer to the fine scale pressure than some given tolerance. We also want to find a total velocity field $\vec{v}_t$ such that $\nabla \cdot \vec{v}_t = q_t$, a property which is sometimes referred to as a *conservative velocity field* for incompressible flow. If we consider the pressure equation (2.13) it is obvious that we cannot in general fulfill Darcy's law $\vec{v}_t = -\mathbf{K}\lambda_t \nabla p$ if the pressure is not solved exactly. For this reason, the exact correspondence between the total velocity and the pressure is typically the sacrifice made when deriving multiscale methods. It is usually considered more important to achieve exact mass-balance, a known first-order principle, than it is to achieve exact fulfillment of the multiphase Darcy's law, which is considered to be empirical.

If we define the fine scale domain as the collection of cells $\{\Omega_i\}_{i=1}^n$, we can also define a coarse grid based on some given partition $\{\bar{\Omega}_j\}_{j=1}^m$ such that each fine cell in Ω belongs to exactly one coarse block in $\bar{\Omega}$. In order to reduce the dimensionality of (4.1), we introduce a coarse pressure $\mathbf{p}_c$ defined on $\bar{\Omega}$ and a numerical prolongation operator that maps pressures defined on the coarse scale to the fine scale $P : \{\bar{\Omega}_j\} \to \{\Omega_i\}$. We also define an analogous map in the opposite direction $R : \{\Omega_i\} \to \{\bar{\Omega}_j\}$. These operators are represented as matrices $\mathbf{P}$ and $\mathbf{R}$ of size $n \times m$ and $m \times n$, respectively. By using the prolongation operator, we can define an approximate fine scale pressure $\mathbf{p}_f$ on Ω from any pressures defined on $\bar{\Omega}$,

$$\mathbf{p}_f = \mathbf{P}\mathbf{p}_c. \tag{4.2}$$

If we insert the approximate $\mathbf{p}_f$ for the fine scale pressure we obtain

$$\mathbf{A}(\mathbf{P}\mathbf{p}_c) = \mathbf{q}. \tag{4.3}$$

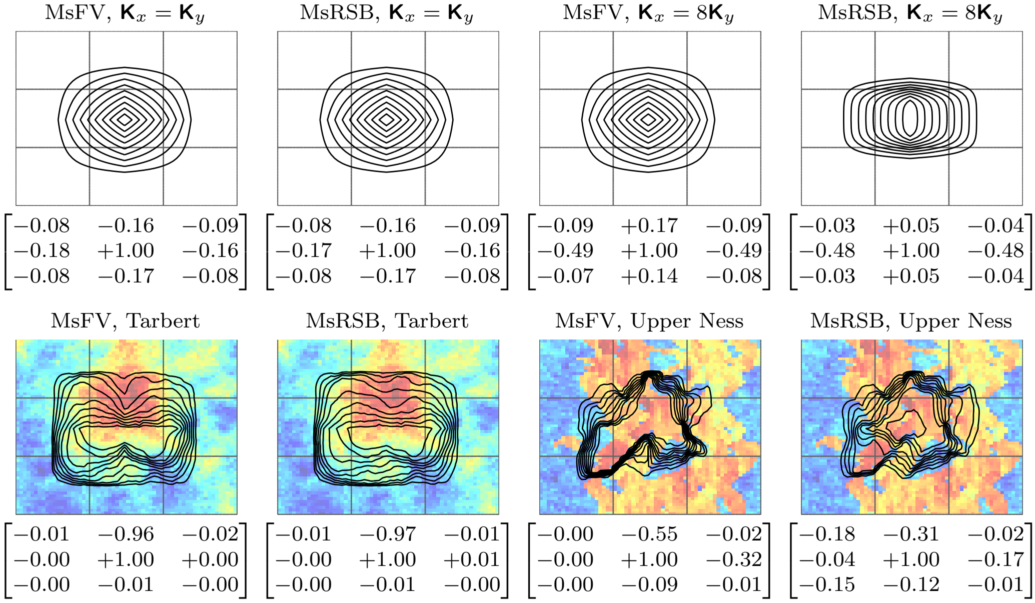
This system now contains more equations than degrees of freedom, and consequently we left multiply with the restriction operator to obtain

$$\mathbf{R}(\mathbf{A}(\mathbf{P}\mathbf{p}_c)) = (\mathbf{RAP})\mathbf{p}_c = \mathbf{A}_c\mathbf{p}_c = \mathbf{R}\mathbf{q} = \mathbf{q}_c \tag{4.4}$$

We now have a square linear system $\mathbf{A}_c\mathbf{p}_c = \mathbf{q}_c$ for the coarse pressure that can be solved. The physical interpretation of this coarse system obviously depends on the choices made for $\mathbf{P}$ and $\mathbf{R}$ which have intentionally not been explained up to this point.

Choice of prolongation and restriction operators

The quality of the multiscale solver obviously depends on the choices made for the restriction and prolongation operator. The prolongation operator represents the multiscale *basis functions*, where column i of $\mathbf{P}$ contains the basis function corresponding to coarse block i. The basis function for a coarse block represents the localized pressure response from unit pressure in the coarse block itself. For the basis functions, there are several options, most of which consist of solving some local flow problem in local domains near the coarse block. One of the primary objectives for this thesis has been to develop new ways of constructing basis functions that are robust with respect to strong heterogeneities and straightforward to implement on fully unstructured grids. The classical MsFV uses a localization assumption and a dual coarse grid to compute basis functions. Figure 4.1, taken from Paper V, demonstrates the isocontour lines of the basis functions for the MsFV and multiscale restriction smoothed basis (MsRSB) methods for four different permeabilities.



MsFV, $\mathbf{K}_x = \mathbf{K}_y$

$$\begin{bmatrix} -0.08 & -0.16 & -0.09 \\ -0.18 & +1.00 & -0.16 \\ -0.08 & -0.17 & -0.08 \end{bmatrix}$$

MsRSB, $\mathbf{K}_x = \mathbf{K}_y$

$$\begin{bmatrix} -0.08 & -0.16 & -0.09 \\ -0.17 & +1.00 & -0.16 \\ -0.08 & -0.17 & -0.08 \end{bmatrix}$$

MsFV, $\mathbf{K}_x = 8\mathbf{K}_y$

$$\begin{bmatrix} -0.09 & +0.17 & -0.09 \\ -0.49 & +1.00 & -0.49 \\ -0.07 & +0.14 & -0.08 \end{bmatrix}$$

MsRSB, $\mathbf{K}_x = 8\mathbf{K}_y$

$$\begin{bmatrix} -0.03 & +0.05 & -0.04 \\ -0.48 & +1.00 & -0.48 \\ -0.03 & +0.05 & -0.04 \end{bmatrix}$$

MsFV, Tarbert

$$\begin{bmatrix} -0.01 & -0.96 & -0.02 \\ -0.00 & +1.00 & +0.00 \\ -0.00 & -0.01 & -0.00 \end{bmatrix}$$

MsRSB, Tarbert

$$\begin{bmatrix} -0.01 & -0.97 & -0.01 \\ -0.00 & +1.00 & +0.01 \\ -0.00 & -0.01 & -0.00 \end{bmatrix}$$

MsFV, Upper Ness

$$\begin{bmatrix} -0.00 & -0.55 & -0.02 \\ -0.00 & +1.00 & -0.32 \\ -0.00 & -0.09 & -0.01 \end{bmatrix}$$

MsRSB, Upper Ness

$$\begin{bmatrix} -0.18 & -0.31 & -0.02 \\ -0.04 & +1.00 & -0.17 \\ -0.15 & -0.12 & -0.01 \end{bmatrix}$$

Figure 4.1: Matrix-dependent interpolation operators for a single coarse block with different types of permeability types. The lognormal and the channelized permeabilities in the lower row are both sampled from the SPE10 dataset. The matrices report the net fluxes into or out of the neighboring coarse blocks induced by a unit pressure differential for a given basis functions. This figure is reproduced from Paper V.

For the restriction operator, there are two options that are commonly used in the literature. The first choice is to use a control-volume restriction operator, which is defined as the characteristic function of each coarse block, and the second is to use a Galerkin-type restriction operator which is the transpose of the prolongation operator. On matrix form, these choices can be written as

$$(\mathbf{R}_{cv})_{ji} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \bar{\Omega}_j, \\ 0, & \text{otherwise,} \end{cases} \qquad \text{or} \qquad \mathbf{R}_G = \mathbf{P}^T.$$

We note that the original formulation of the classical MsFV used the analytical equivalent of the control-volume restriction operator, which can be interpreted as defining the coarse matrix connection strength between degrees of freedom as the estimated flux from the basis functions over the coarse control-volumes. This is conceptually similar to techniques used for transmissibility upscaling based on localized flow problems and is a prerequisite for the flux reconstruction used to obtain a conservative velocity field on the fine grid.

Using the control-volume restriction operator can lead to negative transmissibilities in the presence of strong heterogeneity on the coarse scale, which implies that the multiscale solver will induce flow going from low to high pressures. The resulting multiscale solution may have severe errors, non-monotone pressure and even negative values. For this reason, using a Galerkin-type restriction operator has been favored when pressure stability is a concern, as noted by Wang et al. [122].

Flux reconstruction

Once we have obtained an approximate pressure solution $\mathbf{p}_f$, the velocity field that results from a straightforward application of the discrete Darcy's law will not necessarily be conservative. To obtain conservative fluxes, local Neumann flow problems are solved for the interior of each primal coarse block $\bar{\Omega}_j$ with boundary conditions equal to the flux values used in the construction of the coarse system. Once a *reconstructed pressure* has been found, the fine scale fluxes in the interior of each block are computed from this pressure [44].

Iterative multiscale

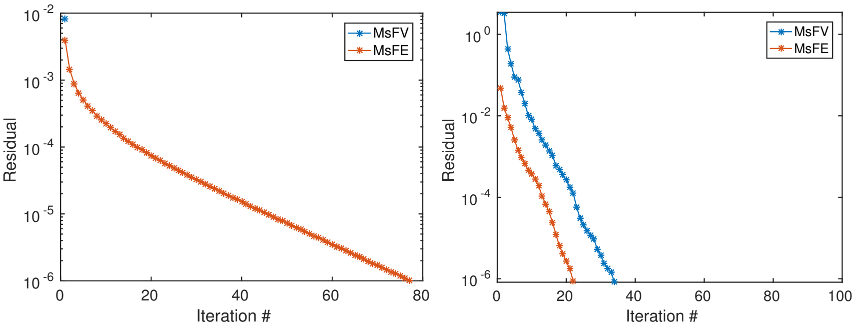
It is known that the multiscale solvers can give high-quality approximations, but for certain applications it is important to systematically reduce error below a given threshold. Iterative multiscale solvers have been introduced for this purpose and have been used to either reduce the residual systematically towards zero, or to remove unacceptable values and non-monotonicities in the solution [33, 122, 76]. To formulate an iterative solver from the multiscale formulation, we proceed much in the same way as when deriving a two-level algebraic multigrid method. We let the solution at step k of the iterative scheme be denoted $\mathbf{x}^k$ to avoid confusion with the pressure used in a temporal discretization, and define the defect $\mathbf{d}$ in the standard manner,

$$\mathbf{d}^k = \mathbf{q} - \mathbf{A}\mathbf{x}^d. \tag{4.5}$$

We also assume that we have some function $S(\mathbf{A}, \mathbf{b})$ which, for a given matrix $\mathbf{A}$ and a right hand side $\mathbf{b}$, performs one or more smoothing iterations. The term smoother refers to any kind of inexpensive iterative solver that efficiently removes high frequency errors from the solution. Examples of possible smoothers include incomplete LU-factorization with zero or low degree of fill-in or standard iterative solvers such as variants of Gauss-Seidel or Jacobi's method. With this in mind, we can define a two-step preconditioner that first removes local error using the smoother and then computes a coarse scale correction,

$$\begin{aligned}
\mathbf{x}^{k+1/2} &= \mathbf{x}^k + S(\mathbf{A}, \mathbf{q} - \mathbf{A}\mathbf{x}^k) = \mathbf{x}^k + S(\mathbf{A}, \mathbf{d}^k) && (4.6)\\
\mathbf{x}^{k+1} &= \mathbf{x}^{k+1/2} + \mathbf{P}\mathbf{A}_c^{-1}\mathbf{R}(\mathbf{q} - \mathbf{A}\mathbf{x}^{k+1/2}) = \mathbf{x}^k + \mathbf{P}\mathbf{A}_c^{-1}\mathbf{R}\mathbf{d}^{k+1/2}. && (4.7)
\end{aligned}$$

After the coarse scale correction has been computed, the solution is again conservative over the coarse control volumes, provided that the control-volume restriction operator is employed. The iterative multiscale solver can be used to get solutions of any accuracy, but there is a significant cost associated with iterating to machine precision since certain medium frequency error modes are not captured. It is also straightforward to use Krylov-based methods to accelerate the solution process if the linear tolerance is strict. In Figure 4.2, the MsFV and MsFE methods are tested on layer 85 of the SPE 10, model 2 dataset. Note that the finite-element version significantly outperforms the control-volume version, which diverges for the unstabilized iterations.



(a) Without GMRES

(b) With GMRES

Figure 4.2: Four different iterative solvers based on the same basis functions. For the heterogenous, channelized permeability distribution sampled from SPE 10, model 2, layer 85, the control-volume restriction operator gives a solver with monotonicity issues that does not converge. By using the method as a preconditioner for GMRES, the convergence rates are improved, but the finite-element restriction operator still significantly outperforms the control-volume version. ILU(0) is used as the second-stage smoother.

Chapter 5

Object oriented framework for experimental programming

5.1 Introduction

Motivation

In the preceding chapters, we have described how the simulation of flow and transport in porous media is done using a wide variety of model equations ranging from incompressible to the compositional. Many of these models are similar to each other in certain aspects, and if we want to develop advanced linear and non-linear solvers or test novel discretization techniques, we need a flexible framework that allows for rapid prototyping and validation.

The flow and transport solvers in the first releases of MRST (see Chapter 2) were limited to incompressible problems with simplified fluid physics, but over time solvers for compressible and miscible flow were introduced. The current set of simulators in MRST that include compressibility and/or miscibility all use automatic differentiation (AD) [87] and this thesis has made importan contributions to this capability. By using a combination of operator overloading, the chain rule and basic rules for derivatives of functions, it is possible to write simulators without manually implementing Jacobians for Newton's method, a process that quickly becomes tedious and time-consuming for complex sets of equations. The current research simulator at Stanford, AD-GPRS, pioneered the use of AD for development of reservoir simulators [119, 124, 129].

Automatic differentiation was originally introduced to MRST in order to solve optimization problems, which resulted in the first generation of AD solvers named `ad-fi` due to the singular focus on fully implicit methods with adjoints [83]. Although widely successful in terms of research productivity, the first generation of AD solvers were victim of their own success. The technology, although very flexible for prototyping, encouraged users to write their own solver stack for each application. This results in many different implementations for what is essentially the same logic. However, there is no obvious reason why a two-phase oil-water model should have a separate implementation of a preconditioned linear solver from the three-phase oil-water-gas model. In order to facilitate the development of novel physical models and mathematical methods without unnecessary code growth and complexity, a new framework based on object orientation was introduced.

The automatic differentiation framework based on object orientation (MRST AD-OO) was designed from the ground up as a part of this thesis to experiment with a variety of physical models, different discretization techniques and solver options, while still retaining all functionality and ease of development from the first generation of solvers. The underlying vision was that by separating the different layers of the simulator into classes and functions with clear interfaces, it would be possible for researchers to develop new simulators and solvers without in-depth knowledge outside their preferred field of expertise.

In this chapter, the structure of the framework is outlined by considering the different components and how they interact during a simulation. An example of how to design a new model and subsequently solve it using Newton's method is also detailed.

Automatic differentiation

Stating and implementing the residual function is usually easy compared to deriving and implementing the Jacobian of the residual function, where differentiation is often carried out by hand and at great length for systems with a large number of active components and equally many partial derivatives. Automatic differentiation is a technique wherein the Jacobian of a function is automatically generated by implementing the differentiation rules for common operators. In conventional programming, a variable will contain one or more values. If the variable is of AD type, however, it will contain both the values of the function in the current point, but also the derivatives of the function with respect to all the primary variables.

5.2 The MRST AD-OO framework

Outline of the framework

The AD-OO framework has a large number of different classes and functions, but the core design includes four main parts. Together, these components make up the entire simulation stack used for simulating a wide variety of models and cases. We will briefly go over the main components in some detail.

Models that contain equations, valid values and discretizations

The most central building block in MRST AD-OO is the model concept. All models are inherited from the virtual base class `PhysicalModel`. A model contains everything necessary to perform a single *step*, where a step is the process by which the model takes an initial state and advances the solution quantities in that state as to get closer to some convergence criterion. If the model represents an implementation of a fully implicit nonlinear equation, a step amounts to the assembly and solution of a linearized set of residual equations with a subsequent update, making the step equivalent to a step of Newton's method. Other models, such as the sequential model we will examine in a later section, contain complex logic with nested solution of different nonlinear systems in order to advance one step.

The model must contain a way to update the solution quantities, often by the way of a Newton-like process. In addition, the model must contain a convergence criterion for the solution quantities, as well as rules for valid values based on the physical constraints of the problem. The framework can be considered *lazy* in that reasonable defaults will be created where they are missing, for instance for functions that update the problem state or a convergence criterion.

The `simulateScheduleAD` function

`simulateScheduleAD` is the main simulator gateway for AD-OO problems. One of few regular functions in the framework, it is used for simulating a full *schedule* consisting of multiple timesteps with different controls during the simulation. It represents the closest thing to a black box simulator in MRST. If we have defined a model, initial problem state and a schedule, we can call the simulator:

```
[wellSols, states, report] = ...
simulateScheduleAD(state0, model, schedule, 'NonLinearSolver', solver)
```

The first input argument is the initial state, which is a struct containing all variables at the beginning of the simulation, typically pressure and saturations for a simple model. The second input is the model, which will be used to advance the state from one time-step to the next, as well as validate that the correct data is present in the initial state. Finally we have the schedule, which is a special struct containing a list of control steps, representing specific points in time where we want know the solutions to our problem, as well as driving forces that specify the wells, boundary conditions and source terms for each step. Note that we can also input a `NonLinearSolver` as an optional input argument, if we are not satisfied with the default configuration, for instance we want to modify the maximum number of iterations used in a step.

Once the simulation is complete, we will get returned an array of `wellSols`, containing the state of the wells (rates, composition, pressures) for each control step, as well as the `states` cell array which contain all solution variables for each step. The optional third output argument, the `report`, holds detailed information about convergence rates, computational cost of each step, number of iterations and any other output produced by the non-linear solver during the solution process.

A class for the solution of non-linear problems

`NonLinearSolver` is used during the simulation to solve the nonlinear problems during each timestep. The class includes management of time-steps, book-keeping of convergence reports and relaxation. For problems where a full schedule is not convenient, the solver can be used as a stand-alone class for solving nonlinear problems. The nonlinear solver relies on the model class to perform iterations and report issues and as such is not a Newton solver by itself. A notable example of an alternative nonlinear solver in MRST is the `TrustRegionSolver` class which implements the generic fluxsearch trust region algorithm described in Paper XIII.

General linear solver class

The `LinearSolverAD` superclass is used to implement new or integrate existing linear solvers. The linear solvers can be problem aware and use information provided by the model in order to improve convergence or automatically configure the solvers. Examples include `BackslashSolverAD`, `GMRES_ILUSolverAD`, `CPRSolverAD` and `MultiscaleVolumeSolverAD`. As fast solvers for large, specialized problems is a known bottleneck in Matlab, it

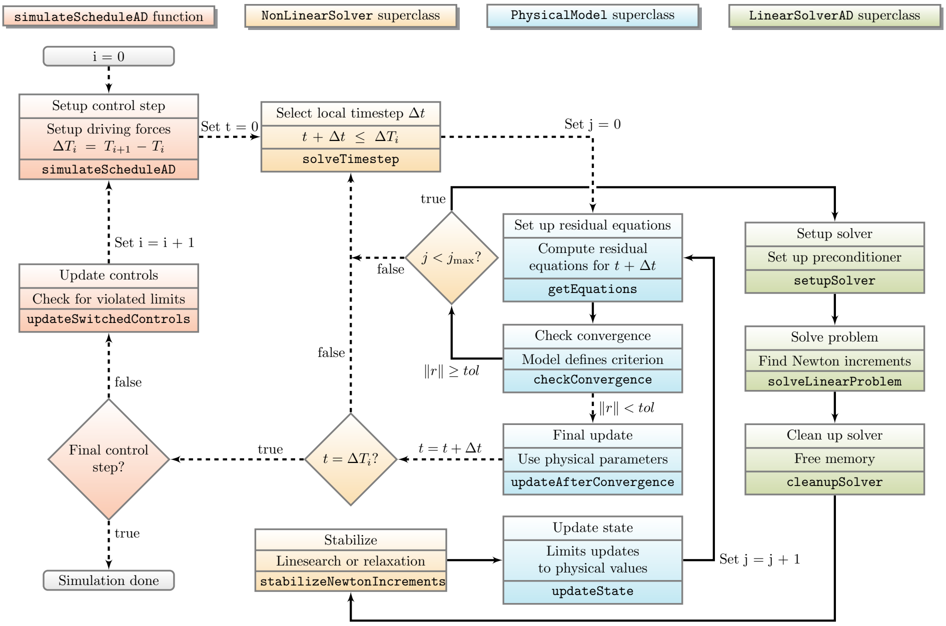


Figure 5.1: Program flow for a fully-implicit simulator based on the AD-framework. Events are colorized by the function or class they belong to, with the relevant member functions highlighted at the bottom of each event. The main solver loop for a single timestep is shown as arrows with solid lines.

is easy to create bindings to external packages, such as the `AGMGSolverAD` which couples to the algebraic multigrid solver by Notay [91].

Figure 5.1 attempts to show how the four components interact via subfunctions during a simulation. In addition to these main components, there exists several smaller components that are more or less integrated into the framework:

- `computeGradientAdjointAD` runs a backward/adjoint [43, 55] simulation in the same manner that `simulateScheduleAD` runs a forward simulation. The solution variables in a backward simulation are the sensitivities of an objective function, giving inexpensive gradients for optimization at the cost of n linear problems for n control steps.

- `computeGradientPerturbationAD` computes gradients using perturbated well controls. The adjoint version is preferable, as it is much less expensive for almost all purposes.

- `LinearizedProblem` implements a class for storing a *linearized prob-*

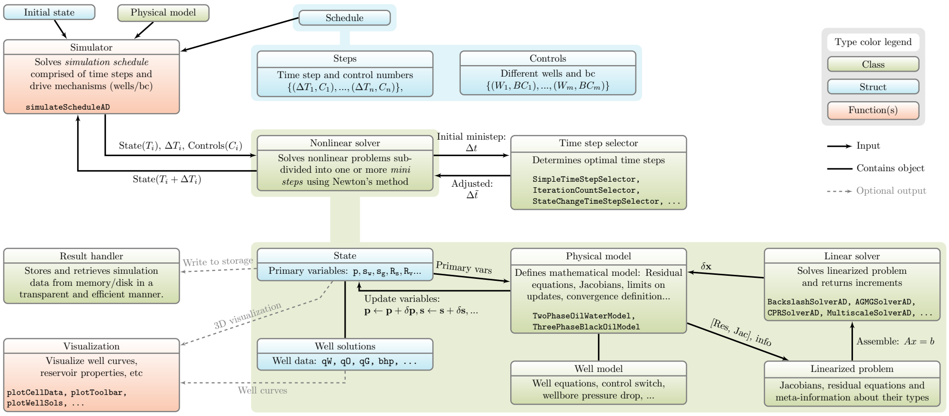

Initial state
Physical model
Schedule
Simulator
Solves *simulation schedule* comprised of time steps and drive mechanisms (wells/bc)
simulateScheduleAD
Steps
Time step and control numbers $\{(\Delta T_1, C_1), ..., (\Delta T_n, C_n)\}$,
Controls
Different wells and bc $\{(W_1, BC_1), ..., (W_m, BC_m)\}$
Type color legend
Class
Struct
Function(s)
Input
Contains object
Optional output
State(T_i), ΔT_i, Controls(C_i)
State($T_i + \Delta T_i$)
Nonlinear solver
Solves nonlinear problems subdivided into one or more *mini steps* using Newton's method
Initial ministep: Δt
Adjusted: $\Delta\tilde{t}$
Time step selector
Determines optimal time steps
SimpleTimeStepSelector, IterationCountSelector, StateChangeTimeStepSelector, ...
Result handler
Stores and retrieves simulation data from memory/disk in a transparent and efficient manner.
Write to storage
State
Primary variables: $\mathtt{p}, \mathtt{s_w}, \mathtt{s_g}, \mathtt{R_s}, \mathtt{R_v}$...
Primary vars
Update variables: $\mathbf{p} \leftarrow \mathbf{p} + \delta\mathbf{p}, \mathbf{s} \leftarrow \mathbf{s} + \delta\mathbf{s}$, ...
Physical model
Defines mathematical model: Residual equations, Jacobians, limits on updates, convergence definition...
TwoPhaseOilWaterModel, ThreePhaseBlackOilModel
$\delta\mathbf{x}$
Linear solver
Solves linearized problem and returns increments
BackslashSolverAD, AGMGSolverAD, CPRSolverAD, MultiscaleSolverAD, ...
[Res, Jac], info
Assemble: $Ax = b$
3D visualization
Visualization
Visualize well curves, reservoir properties, etc
plotCellData, plotToolbar, plotWellSols, ...
Well solutions
Well data: qW, qO, qG, bhp, ...
Well curves
Well model
Well equations, control switch, wellbore pressure drop, ...
Linearized problem
Jacobians, residual equations and meta-information about their types

Figure 5.2: Relationships between different components of the AD-OO framework. This figure gives a more high-level overview of the program flow than 5.1 as it demonstrates the steps taken to simulate a whole *schedule* comprised of many different steps and controls.

lem, namely a set of equations, knowledge about what the variables are and the state that produced them. The class is also used to assemble linear systems from the linearized equations stored in AD-format.

- `WellModel` represents wells and essentially implements Peaceman's model [95]. In addition, the class handles changing well-controls during the simulation due to predefined limits being exceeded, e.g. maximum bottom-hole pressure.

- `SimpleTimeStepSelector` is a base class for selecting time-steps. The base class simply features minimum and maximum time-steps, but subclasses (`IterationCountStepSelector` or `StateChangeTimeStepSelector`) implement more advanced heuristics for timesteps. The time-step selector can be attached to any `NonLinearSolver` subclass.

- `ResultHandler` allows for transparent storage of simulation output. Large simulations can be written directly to disk to avoid filling up the memory during a simulation, or to obtain partial results when a simulator is running.

5.3 Building a new model with AD-OO

All new models in the AD framework start out by subclassing a `PhysicalModel`, either directly or via some other subclass that has parts we want to use.

The `PhysicalModel` superclass can be thought of as a collection of all the things that define a specific model. The textbook definition of a model may not be enough to obtain a working numerical model: A model should also incorporate the know-how on how to update the variables to ensure good convergence, what form of the equations is best suitable for the numerical methods, and so on. All this knowledge can be incorporated into the model class. To implement a new fully-implicit model, typically only two things need to be done:

- The first is the residual form of the model equations, implemented as the `getEquations` member function.
- The second is the update function which translates the increments from the Newton solver into a updated state struct, implemented as the *updateState* function.

Fortunately, the *PhysicalModel* contains several member functions that make it easy to do this. We will begin by looking at a simple textbook model that can be used in this manner before showing more advanced features.

Rosenbrock Model Problem

The Rosenbrock model problem is a classical optimization test problem that can be solved by Newton's method. We have the objective function,

$$f(x,y) = (a-x)^2 + b\left(y-x^2\right)^2, \qquad a,b \in \{z \in \mathbb{R} : z > 0\}, \tag{5.1}$$

and we want to obtain a minimum. It is straightforward to see that the global minimum of this function is (a, a^2). We want to see if the Newton solver is up to the task by implementing this optimization problem as a subclass of `PhysicalModel`. It is well known that a requirement for a global minimum is that

$$\nabla f(x,y) = 0, \tag{5.2}$$

and from this we obtain our two model equations

$$\frac{\partial f}{\partial x} = -2(a-x) - 4bx(y-x^2), \qquad \frac{\partial f}{\partial y} = 2b(y-x^2). \tag{5.3}$$

The first thing we do is to subclass `PhysicalModel` directly, adding properties for the a and b coefficients. We do not add x and y to properties, as these are solution variables during the simulation and thus parts of the *state* and not the model itself. We also add a very simple class constructor.

```
classdef RosenbrockModel < PhysicalModel
    properties
        a, b
    end
    methods
        function model = RosenbrockModel(a, b, varargin)
            % Class constructor for tne Rosenbrock model
            % Call parent class to get default values. The grid value is
            % empty, as we do not have an explicit grid for this problem.
            model = model@PhysicalModel([]);
            [model.a, model.b] = deal(a, b);
            % Allow keyword arguments to override any defaults in the base class
            model = merge_options(model, varargin{:});
        end
    end
end
```

The class constructor contains a bit of extra code to first call the parent class and merging together keyword arguments with the defaults. This allows us to for example adjust the default nonlinear tolerance measured in the inf-norm, and enable verbose output, both of which are inherited from `PhysicalModel`.

```
model = RosenbrockModel(1, 100, 'nonlinearTolerance', 1e-12, 'verbose', true)
```

The properties of the model is output to the command window,

```
RosenbrockModel with properties:
a: 1
b: 100
operators: []
nonlinearTolerance: 1.0000e-12
G: []
verbose: 1
```

Now that we have our base class up and running, it is time to add the knowledge of the residual equations to the model. We add another method for this purpose:

```
function [problem, state] = getEquations(model, state0, state, drivingForces, dt)
    % Get X and Y
    [x, y] = model.getProps(state, 'x', 'y');
    % Convert to AD objects
    [x, y] = initVariablesADI(x, y);
    % Gradient is implemented as two equations
    dfdx = -2.*(model.a-x) - 4.*model.b.*(y-x.^2).*x;
    dfdy = 2.*model.b.*(y-x.^2);
    % Combine into a LinearizedProblem
    problem = LinearizedProblem({dfdx, dfdy}, {'scalar', 'scalar'}, ...
                                {'df/dx', 'df/dy'}, {'x', 'y'}, ...
                                state);
end
```

When calling the *LinearizedProblem* constructor, we add some extra information about our equations, namely that they are scalars, their names are 'df/dx' and 'df/dy', and that the solution variables are x and y. For this example the equation names and types will only be used for printing the convergence history, but as these are passed onto the linear solver they can in principle be used to speed up the solution process.

One may also note that we do not access the state struct directly, but rather use member functions inherited from `PhysicalModel`. Although it is still possible to access the state directly, this approach has the advantage of defining how model data is stored canonically, making it easy for the solver to determine when fields are accessed inconsistently. The next step is then to add the canonical mapping for properties `x` and `y`, which is taken care of by the model member function `getVariableField`. This function takes in a name and produces the state field name `fn` and the column `index` of the data. In this case we simply want to find `x` and `y` at `state.x` and `state.y` respectively.

```
function [fn, index] = getVariableField(model, name)
    % fn is the fieldname, index is the second subsref index.
    switch(lower(name))
    case 'y'
        fn = 'y';
        index = ':';
    case 'x'
        fn = 'x';
        index = ':';
    otherwise
        % Basic values are known to the base class
        [fn, index] = getVariableField@PhysicalModel(model, name);
    end
end
```

This implementation simply tells the AD framework how to access the x and y properties:

```
model.getProp(state, 'x')
% Is translated into
state.x(:, :)
```

Note that for values that are not scalar per cell, the index can be used to access different elements. For instance, consider a `ThreePhaseBlackOilModel`, which has the field `.s` with individual columns for water, oil and gas:

```
model.getProp(state, 'sg')
% Is translated into the third column, as the gas phase is the third phase
state.s(:, 3)
% To get all saturations, simply use s by itself
model.getProp(state, 's')
state.s(:, :)
```

If this seems needlessly bureaucratic, do not worry. It is still possible to index directly into the state for simple problems, but this approach has the advantage of making it easy to write general code that handles for instance one and three phase models with the same code. Fortunately, this also makes the update function for implicit schemes automatically generated.

We now have everything we need to solve the nonlinear problem! To test the function we simply create an instance of the default non-linear solver and call the `solveTimestep` function with our initial guess, a dummy time-step and the model itself,

```
% Create Non-Linear solver
solver = NonLinearSolver();
% Initial guess
state0 = struct('x', -0.5, 'y', 4);
state = solver.solveTimestep(state0, 0, model);
fprintf('Final values: x = %f, y = %f\n', state.x, state.y)
```

Because we instantiated the class with the verbose flag, it will print output every time the residuals are checked against the convergence criterion. The output to terminal is:

```
==========================================
| It # | df/dx (scalar) | df/dy (scalar) |
==========================================
|    1 | 7.47e+02       | 7.50e+02       |
|    2 | 3.00e+00       | 8.02e-04       |
|    3 | 9.00e+02       | 4.50e+02       |
|    4 | 2.40e-03       | 1.42e-09       |
|    5 | 5.77e-04       | 2.89e-04       |
|    6 | 4.35e-14       | 2.22e-14       |
==========================================
Final values: x = 1.000000, y = 1.000000
```

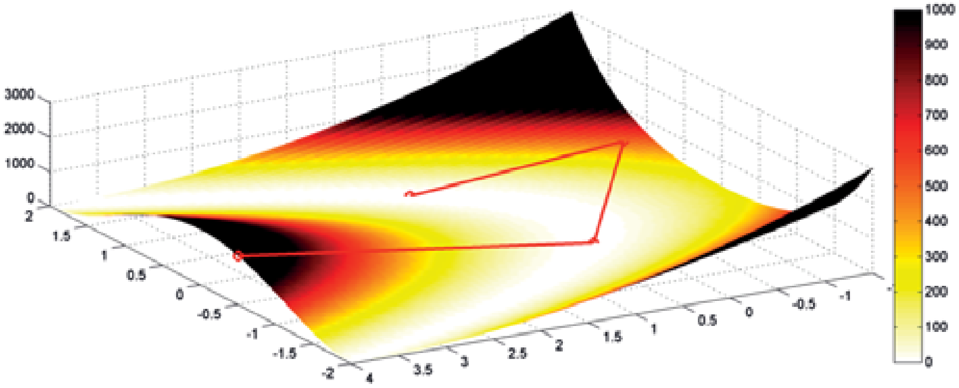


Figure 5.3: The path (red) taken by the nonlinear solver on the Rosenbrock test problem, shown superimposed over the objective function response surface. By setting the colormap to a variant of hot it becomes apparent why this function is sometimes referred to as the banana test function.

As can be seen from the output, the solution converges to the global minimum rapidly. The path taken by the nonlinear solver has been plotted in Figure 5.3. With only a few lines of code, we implemented and solved a new nonlinear problem and we immediately have access to the full suite of linear solvers available to MRST if we want to solve larger problems.

A model for single-phase compressible flow

For our second example, we will implement a model for single-phase compressible flow. The governing equation for this model is the compressible pressure equation, on semi-discrete form,

$$\frac{\phi}{\Delta t}(\rho^{n+1} - \rho^{n}) + \nabla \cdot \vec{v} - q = 0, \tag{5.4}$$

where the velocity field is given by Darcy's law,

$$\vec{v} = -\frac{1}{\mu}\mathbf{K}\nabla p. \tag{5.5}$$

This equation is similar to the incompressible pressure equation (2.13), but the addition of a density ρ that depends on the pressure makes the problem compressible, and consequently we have a nonlinear parabolic equation instead of a linear elliptic equation.

To implement this as a model in AD-OO, we can proceed as we would for the Rosenbrock model problem: We create a subclass of the `ReservoirModel`, which implements *discrete operators* that contain the discretizations described in Chapter 3. The complete implementation of the single-phase model can then be written as less than 30 lines of actual code:

```
classdef CompressibleSinglePhaseModel < ReservoirModel
    methods
    function model = CompressibleSinglePhaseModel(G, rock, fluid, varargin)
        % Default constructor required by Matlab
        model = model@ReservoirModel(G, rock, fluid, varargin{:});
    end
    function [problem, state] = getEquations(model, state0, state, dt,...
     drivingForces, varargin)
        p = model.getProp(state, 'pressure'); % Current pressure
        p0 = model.getProp(state0, 'pressure'); % Previous pressure
        p = initVariablesADI(p); % Convert to AD objects
        f = model.fluid; % Evaluate properties
        rho = f.rho(p); % Current density
        rho0 = f.rho(p0); % Density at the previous time-step
        s = model.operators; % Short hand for operators
        vT = -(1./f.mu).*s.T.*s.Grad(p); % Darcy's law
        eq = (s.pv./dt).*(rho - rho0) + s.Div(vT); % Pressure equation
         if ~isempty(drivingForces.src)
            % If provided, add in source terms to  cells.  Sources are given as
            % volumetric rates at surface conditions, convert to mass.
            srcCells = drivingForces.src.cell;
            q = f.rhoS*drivingForces.src.rate;
            eq(srcCells) = eq(srcCells) - q;
        end
        % Assemble problem: The pressure equation is defined per cell, with pressure
        % as the primary variable
        problem = LinearizedProblem({eq}, {'cell'},{'pressureEquation'}, ...
                                        {'pressure'},state);
        end
    end
end
```

We did not have to implement additional canonical mappings, as the base class has knowledge of most standard properties, including the pressure. Details of the discretization are abstracted away, and the final implemented equations are very similar to the semi-discrete form in (5.4).

We can define a simple test problem for our new model. We define a 10×10 discretization of the unit square. The fluid model has density that depends on the pressure,

$$\rho(p) = \rho_0 \, e^{(p-p_0)c}, \tag{5.6}$$

where ρ_0 is the density at the reference pressure p_0 and c is the compressibility constant. The full script for setting up, simulating and plotting the injection of one pore volume over 30 days is 20 lines of code when comments are excluded:

```
% 10 by 10 grid discretizing the unit square
G = computeGeometry(cartGrid([10, 10], [1, 1]));
rock = makeRock(G, 50*milli*darcy, 0.5); % Uniform rock
T = 30*day; % Simulate 30 days
p0 = 250*barsa; % Initial pressure
rho0 = 500*kilogram/meter^3; % Initial density
fluid = struct('rho', @(p) rho0*exp((p - p0)*1e-3/barsa),... % Density
               'mu', 1*centi*poise, ... % Constant viscosity
               'rhoS', rho0); % Surface density for sources
rate = sum(poreVolume(G, rock))/T; % Inject one pore volume in total
% Inject in south-west corner, produce in north-east
src = addSource([], 1, rate);
src = addSource(src, G.cells.num, -rate);
model = CompressibleSinglePhaseModel(G, rock, fluid); % Set up model
state0 = initResSol(G, p0); % Initial state
nstep = 10; % Request output every 1/10th of time period
dt = repmat(T/nstep, nstep, 1);
schedule = simpleSchedule(dt, 'src', src);
[~, states] = simulateScheduleAD(state0, model, schedule); % Solve problem
% Plot the pressure after the final time-step in bar
figure; plotCellData(G, states{end}.pressure/barsa); axis equal tight; colorbar
```

The plot produced by this script is reproduced in Figure 5.4 where we can see the expected radial flow pattern around each source term. Since the model itself is derived from the `ReservoirModel` base class, the new model is immediately applicable to any of the grids available to MRST.

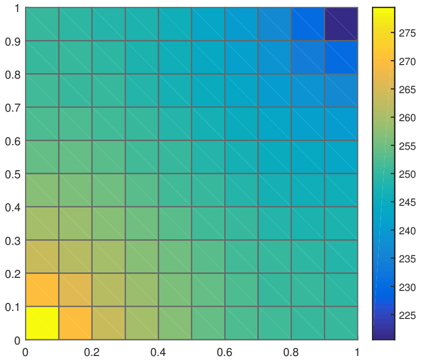


Figure 5.4: The pressure after the simulation of a small test problem using the compressible single-phase solver developed using AD-OO.

5.4 Advanced extensions

We have seen that the definition of new nonlinear problems with Newton solvers is straightforward. We will consider two other examples where nested models are used to implement more advanced solution strategies.

Different model and their relationships

To first give some indication on how the different solvers are represented in the AD-OO framework, consider the inheritance diagram in Figure 5.5. The three different mathematical models of two-phase immiscible flow, three-phase black-oil and compositional are shown in the same figure. For each model, a fully-implicit discretization is considered as the base class, which is then extended to pressure and transport models that can be combined using the generic sequential model. We also see that the compositional models depend require an equation of state model to be constructed. The default choices for primary variables for each model is also included, highlighting that there is significant overlap between models and submodels in terms of functional dependencies and consequently update functions.

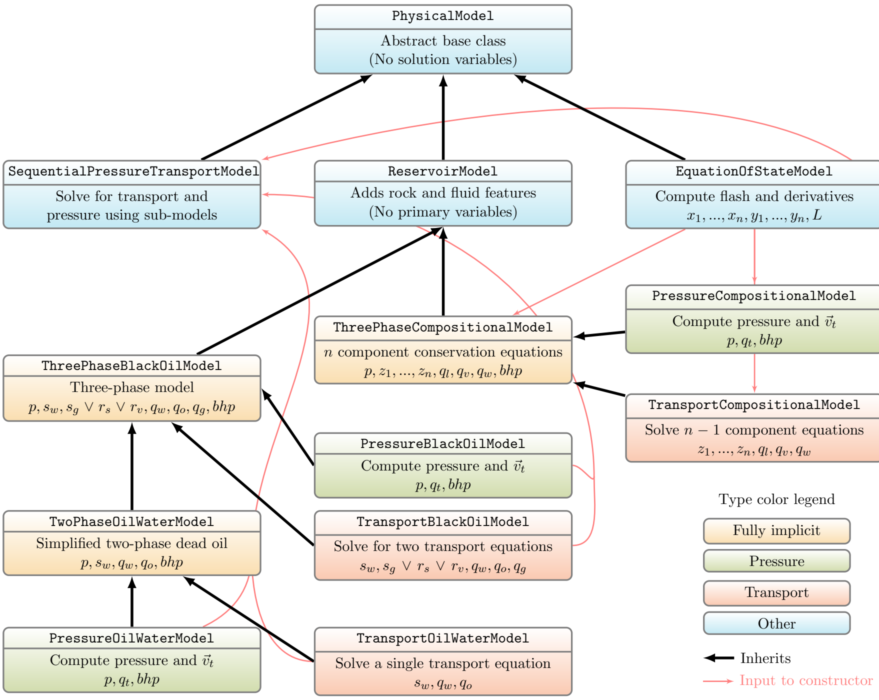

PhysicalModel
Abstract base class
(No solution variables)
SequentialPressureTransportModel
Solve for transport and pressure using sub-models
ReservoirModel
Adds rock and fluid features
(No primary variables)
EquationOfStateModel
Compute flash and derivatives
$x_1, ..., x_n, y_1, ..., y_n, L$
PressureCompositionalModel
Compute pressure and $\vec{v}_t$
p, q_t, bhp
ThreePhaseCompositionalModel
n component conservation equations
$p, z_1, ..., z_n, q_l, q_v, q_w, bhp$
ThreePhaseBlackOilModel
Three-phase model
$p, s_w, s_g \vee r_s \vee r_v, q_w, q_o, q_g, bhp$
TransportCompositionalModel
Solve $n-1$ component equations
$z_1, ..., z_n, q_l, q_v, q_w$
PressureBlackOilModel
Compute pressure and $\vec{v}_t$
p, q_t, bhp
TransportBlackOilModel
Solve for two transport equations
$s_w, s_g \vee r_s \vee r_v, q_w, q_o, q_g$
TwoPhaseOilWaterModel
Simplified two-phase dead oil
p, s_w, q_w, q_o, bhp
PressureOilWaterModel
Compute pressure and $\vec{v}_t$
p, q_t, bhp
TransportOilWaterModel
Solve a single transport equation
s_w, q_w, q_o
Type color legend
Fully implicit
Pressure
Transport
Other
Inherits
Input to constructor

Figure 5.5: Three different flow models (two-phase compressible, three-phase black-oil and compositional) solved using fully-implicit and sequential implicit temporal discretizations.

Sequential solvers

To implement a model for the coupled pressure and transport equations posed in sequential form means that one first has to develop a model for the pressure equation (2.13), and another model for the transport equation on fractional flow form (2.14). The two models can then be combined using the constructor of the `SequentialPressureTransportModel` class, which allows for the combination of different pressure and transport discretizations into a coupled scheme. As either model can have distinct linear and nonlinear solvers, the framework offers a great deal of flexibility with regards to choices for solution strategy.

The `SequentialPressureTransportModel` has a specialized `stepFunction` member function that first solves the pressure, checks for convergence and then proceeds to solve the transport. The relevant lines of code in the class definition are short and easy to read if one is familiar with the scheme,

```
psolver = model.pressureNonLinearSolver;
tsolver = model.transportNonLinearSolver;
% Get the forces used in the step
forceArg = getDrivingForces(model.pressureModel, drivingForces);
% First, solve the pressure using the pressure nonlinear solver
[state, pReport] = psolver.solveTimestep(state0, dt, model.pressureModel,...
                'initialGuess', state,forceArg{:});
if pReport.Converged
    % If pressure converged, we proceed to solve the transport
    [state, transportReport] = ...
        tsolver.solveTimestep(state0, dt, model.transportModel,...
                    'initialGuess', state, ...
                    forceArg{:});
    transport_ok = transportReport.Converged;
else
    transport_ok = false;
    transportReport = [];
end
```

Since each model can have a corresponding nonlinear solver, the pressure and transport can use different timesteps, solution strategies and convergence criterion. There is also an outer nonlinear solver which can manage the *coupled* system of equations, if convergence to the fully-implicit solution is required.

Compositional models with equation of state

Another example of a model which can be decomposed into different parts is the compositional model. The phase behavior and fluid properties are predicted by the equation of state as seen in Chapter 2, but these calculations are not only useful in a flow simulator. For this reason, the equation of state is implemented as a separate model which can be passed to the `NonLinearSolver` to solve the flash equations using either Newton's method or successive substitution.

When we define a compositional model, we can simply pass in the equation of state class instance to the constructor for the flow model,

```
f = CoolPropsCompositionalFluid({'Methane', 'CarbonDioxide'});
EOS = EquationOfStateModel(G, f);
model = ThreePhaseCompositionalModel(G, rock, fluid, EOS);
```

and the compositional flow model will then use the `EquationOfState` class to calculate flash and properties. Since the compositional model relies on a few functions to obtain required quantities from the equation of state, it is easy to insert a different model for the phase behavior with no modification to the flow solver.

Chapter 6

Summary of Papers

The second part of the thesis contains the different papers that have been written as a part of the doctoral studies. Broadly speaking, the papers can be categorized into five themes:

- Papers I, III, IV and V are concerned with the development of *robust and accurate multiscale methods* for unstructured grids. The model equations in these papers are primarily incompressible flow, where the goal is to accurately resolve flow on all scales for problems with challenging geometry and petrophysical properties. Paper I focuses on the extension of the classical MsFV-method to stratigraphic and unstructured grids and illustrates some of the limitations of the localization assumption for such problems. Paper III develops a two-point multiscale method (MsTPFA) that is more robust than the classical MsFV and easily extended to unstructured grids. Paper IV compares different multiscale and other two-level methods as preconditioners in order to assess their relative quality. Paper V introduces the multiscale restriction-smoothed basis method (MsRSB) that is easy to implement for general grids and is robust and accurate for problems with strong heterogeneity.

- In papers VIII, IX, X, XIII and XIV the new MsRSB method is extended to different applications with significant physical complexity. Paper VIII extends the method to compressible black-oil problems. This black-oil capability is then extended to an EOR method in the form of polymer flooding with non-Newtonian fluids in Paper IX. Paper X applies MsRSB to fractured porous media discretized using the embedded fracture approach. Paper XIII details the implementation and validation of MsRSB in a commercial multiscale simulator. Finally, we consider compositional problems in paper XIV.

- Papers II and VI concern the use of *flow diagnostics* that uses reduced physics assumptions to compute fast estimates and other complimentary information that may be expensive or difficult to obtain with conventional multiphase simulation. Flow diagnostics can give volumetric partitions of the simulation domain into different flow regions, measures of dynamic heterogeneity (variation in length of flow paths) and estimates of displacement and sweep efficiency. Paper II details the diagnostics implementation in MRST and demonstrates the use of flow diagnostics with an adjoint formulation for the optimization of well rates and placement. Paper VI demonstrates the use of approximate multiscale solvers to compute diagnostics efficiently, including several examples with model ranking based on uncertain fault multipliers.

- Papers VII and IX detail the use of MRST as a prototyping platform for the nonlinear simulation of flow and transport in porous media. An integral component of this capability is the object oriented framework for writing new solvers based on automatic differentiation. Paper VII details the general capability added to MRST with AD-based solvers. Paper IX go into greater detail on the specific application of polymer flooding, which also serves as an introduction to the class-based hierarchy of solvers used in other papers of the thesis.

- Paper XIII details the development of a modified Newton solver for the transport equations that gives greatly improved convergence rates for three-phase flow with gravity.

In the following pages, motivation and assessment are given for each paper. The actual papers are included in Part II of the thesis.

Paper I

The Multiscale Finite Volume Method on Stratigraphic Grids

Olav Møyner and Knut-Andreas Lie

SPE Journal, volume 19, issue 5, pp. 816-831, 2014
DOI: 10.2118/163649-PA

The first paper of the thesis concerns the extension of the multiscale finite-volume (MsFV) [44] method to unstructured grids. The primary focus of this work was to apply the MsFV-method to stratigraphic corner-point grids with unstructured features. When the paper was written, the MsFV-method was considered to be a promising candidate for practical simulation workflows, in part due to adaptivity [45, 64], error-control in the form of iterative schemes [33] and demonstrations of idealized nonlinear fluid physics [72, 73, 63, 56]. The MsFV-method was originally formulated as a series of analytical problems, which limited the method to structured grids, but a few key papers had reformulated these problems in operator form [90, 126, 74, 75], making it feasible to consider the method for more complex grids.

The extension to unstructured grids was not straightforward, however, as the operator formulation merely poses the problem of basis construction algebraically, given that a wire-basket dual grid ordering of the grid cells can be found. The major obstacle is therefore to obtain a dual-grid which is suitable for localized problems for unstructured grids. In this paper, we consider a triangulation-based approach to construct the dual-grid from a pre-existing primal partition of the domain. We show that it is possible to construct basis functions for a wide range of (semi-)unstructured fine grids, including Voronoi diagrams, prismatic grids and both field and bedding-scale corner-point grids.

It was previously well-known that the MsFV-method would suffer from issues with over- and undershoots in the pressure field for problems with high contrast [50]. As the errors in the solution appear from the localization assumption used to construct basis functions, this problem was also the part of the driving force behind the iterative MsFV formulated by Hajibeygi et al. [33]. Unfortunately, the over- and undershoots that were restricted to a relatively small amount of grid cells in 2D will be much more significant in 3D. In the paper, we verify that for certain grids, severe monotonicity problems are observed when the contrast becomes larger. Faults, which result in the sharp transition between different rock-types over the displacement, are especially problematic, as the localization assumption is severely violated when localized problems pass through multiple sands with different properties.

The conclusion of the paper was therefore a mixed success: While we were capable of producing semi-automated coarsening for a wide variety of grids, the underlying localization assumption for the MsFV-method would lead to unusable results. For this reason, we did not pursue dual grid generation any further and instead shifted focus to robust approaches for basis generation that were also applicable to general grids.

Paper II

The Application of Flow Diagnostics for Reservoir Management

Olav Møyner, Stein Krogstad and Knut-Andreas Lie

SPE Journal, volume 20, issue 2, pp. 306–323, 2014
DOI: 10.2118/171557-PA

The second paper concerns the use of flow diagnostics for reservoir management. Flow diagnostics is a post-processing technique where output from a simulation can be used to understand connections in the reservoir, as well as properties of the flow field that are not directly obtainable from standard simulators. The paper describes the discretization of the time-of-flight equation and the closely related tracer equations, as well as derived quantities that are useful to gain an understanding of the recovery processes.

Traditionally done using streamline methods [38, 40], a paper by Shahvali et al. [102] built upon earlier work by SINTEF on discretizing time-of-flight using Eulerian methods [11, 85] demonstrated that comparable properties could be found for complex grids without implementing streamlines tracing for general grids, which is known to be challenging. In our paper, we present the work on flow-diagnostics done in MRST, including open source tools for visualization and ranking of models using standard measures [104, 42]. The functionality is extended with an adjoint formulation, which enables the use of flow diagnostics for optimization. Since the derived quantities used in flow diagnostics are inexpensive to compute compared to a fully-featured flow simulation, we can rapidly optimize well rates and placements for waterflooding. We demonstrate that the reduced models agree with the fully featured multiphase flow simulations, indicating that flow diagnostics can be a suitable alternative or accelerator to conventional optimization processes. A more recent paper [54] shows that similar reduced models can be used to accurately characterize the more complex recovery process of polymer flooding.

Paper III

A Multiscale Two-Point Flux-Approximation Method

Olav Møyner and Knut-Andreas Lie

Journal of Computational Physics, volume 275, pp. 273–293, 2014
DOI: 10.1016/j.jcp.2014.07.003

Following the issues with the MsFV for unstructured grids and strong permeability contrast, the second paper in this thesis on multiscale methods formulates the multiscale two-point flux approximation (MsTPFA) method. The method is based on the same cell-centered, finite-volume discretization as the MsFV-method, but overcomes the dual grid issues by using a different localization assumption. While the classical MsFV uses a set of successive boundary conditions going from one to two, to three dimensions, we use a simpler form that solves a single flow problem per coarse interface. The problems should be familiar from transmissibility upscaling [18, 97] and the MsMFEM method [1, 2], where such problems have been used to produce robust coarse scale discretizations.

The key innovation of this paper is the combination of the localized solutions with partition-of-unity functions for general grids. These functions are based on localized tracer problems using the discretization techniques from Paper II. For each coarse block, tracer problems subdivide the domain into patches and naturally associates the pressures response of a single coarse face solution with a subset of the fine cells. Using partition-of-unity to combine different analytical functions has also been used with sucess in the extended finite-element method [30].

The numerical results in the paper indicate that the method overcomes the issues with unstructured grids faced by the classical MsFV-method. Numerical tests indicate that the method is more robust, can use general grids and converges rapidly for high-contrast media. Since the local problems can be defined for any coarse grid, we also demonstrate that arbitrary levels of coarsening can be achieved for structurally complex models. We do observe that the error is somewhat higher for problems where the MsFV-method does not suffer from monotonicity issues, however, due to the combination of local solutions not always producing a smooth prolongation operator. In this sense the MsTPFA-method trades accuracy for robustness and generality with respect to grids. Unfortunately, the approach with both localized problems and construction of specific partition-of-unity functions makes the method complex to implement. Eventually, the MsTPFA-method was superseded by the MsRSB-method, which is both more accurate and simpler to implement. One obvious extension of this work is the development of a MPFA-type multiscale method using the same techniques. With a suitable choice of partition-of-unity functions and local solutions, there may be several possible choices that are analogues of fine-scale discretizations.

Paper IV

Construction of Multiscale Preconditioners on Stratigraphic Grids

Olav Møyner

In proceedings of the 14th European Conference on the Mathematics of Oil Recovery, 2014, Catania, Sicily, Italy.
DOI: 10.3997/2214-4609.20141775

The iterative version of the MsFV, referred to as i-MsFV or algebraic multiscale solver (AMS) when posed in algebraic form has been the subject of a number of different papers [127, 122, 76] where choices for iterative strategies using the dual grid basis functions was explored. In 2012, a paper was published by Zhou and Tchelepi [127] that indicated the poor iterative performance of MsFV for certain high-contrast models as rooted in the control-volume integration used to construct coarse scale systems. As this behavior was linked with the monotonicity issues of the method especially prevalent on unstructured grids, we set out to investigate if different choices of basis functions could also mitigate the poor convergence behavior.

Through systematic tests, basis functions based on dual-grids, the MsTPFA basis and basis functions computed using restricted smoothing (MsRSB) were benchmarked against constant and linear interpolation on a wide array of test cases. Although full details of the method was not given in this paper, this is the first paper where the MsRSB basis functions appear. The idea of this method was simple. Start with a constant value inside each coarse block and use a Jacobi iteration to reduce local error. To keep localization, each basis function is not allowed to grow outside a predefined support region, and whenever this happened, the corresponding mass outside the support region was removed and distributed to other basis functions to ensure partition of unity. Where possible, the paper also compares basis functions computed on structured coarse grids with unstructured coarse grids produced by the Metis graph-partitioning package [79] applied to the transmissibilities of the fine-scale problems.

The results were somewhat surprising. For many cases, MsFV was outperformed by the trivial constant basis functions due to severe monotonicity issues. The preliminary implementation of the MsRSB performed very well for most test-cases, with MsTPFA giving comparable, albeit slightly worse performance than MsRSB for several cases. The difference between the different multiscale solvers was significant, but not to the degree as the effect of adapting the coarse grids to the permeability field. With the same number of degrees of freedom, the adaptive grids impacted the convergence rate more than the choice of basis functions.

From these results, we concluded that grid adaption was essential to a successful multiscale method. We remark that, although not in an iterative framework, similar observations has been made for the MsMFE-method [1, 2, 5] and for algebraic multigrid methods [108]. It also became apparent that the iterative multiscale formulation would not necessarily outperform algebraic multigrid methods as exact linear solvers, which have been developed to systematically reduce all components of solution error efficiently.

Paper V

A Multiscale Restriction-Smoothed Basis Method for High Contrast Porous Media Represented on Unstructured Grids

Olav Møyner and Knut-Andreas Lie

Journal of Computational Physics, volume 304, pp. 46-71, 2016
DOI: 10.1016/j.jcp.2014.07.003

After extended unpublished numerical tests, the standard way of constructing basis functions based on localized flow problems did not seem attractive. The work on creating a dual grid for the MsFV that would be robust to strong heterogeneity and faults was eventually not continued, as it appeared that the requirements on the grid in order for the localization assumption to be reasonable were too restrictive to support fully automated gridding for general grids. The MsTPFA-method required non-trivial data structures and pre-processing steps to get partition-of-unity functions that were not considered feasible by the industrial partner.

The MsRSB-method was developed as a result of an extensive study of multigrid theory where the objective was to consider existing techniques for constructing prolongation operator for general grids. The MsRSB method is inspired by the technique used in smoothed agglomeration multigrid [113, 115, 114] where a single pass of a smoother restricted to a connection subset is applied directly to a constant initial guess to get a *locally smooth* prolongation operator. The MsRSB method is defined by a support region consisting of a number of fine cells close to a coarse block, which is then used to localize an iterative process whereby local errors are efficiently removed and basis functions that are perfectly adapted to the features of the underlying medium are obtained. As the requirements on the underlying coarse grid are quite simple, the method is straightforward to implement on fully unstructured grids. With no required localization assumption, the method appears *very* robust to strong contrasts and large aspect ratios. More recently, Lukyanov and Vuik [71] noted that there exist connections between the MsRSB-method and deflation-based meshless methods.

The paper also presents a technique for adaptive control-volume integration based on the isosurfaces of the basis functions that allows the MsFV to avoid most monotonicity issues. This technique, originally intended as a part of an unpublished dual grid algorithm, works well enough to resolve the extreme contrasts of the full SPE 10 model 2 problem for the MsFV, but even with this fix the MsRSB was more accurate. As a result, this is the last paper in the thesis where the dual-grid basis functions are considered. Note that the MsRSB was originally published as Møyner and Lie [82], a paper which was later split into two parts and extended into Papers V and VII. The original conference paper was for this reason omitted from the thesis, but the placement of Paper V is therefore earlier than the publication date would otherwise imply.

Paper VI

Application of Flow Diagnostics and Multiscale Methods for Reservoir Management

Knut-Andreas Lie, Olav Møyner and Stein Krogstad
In proceedings of the 2015 SPE Reservoir Simulation Symposium, Houston, Texas, USA
DOI: 10.2118/173306-MS

Previous work on multiscale methods have for the most part focused on the flow prediction for a specific fine-scale problem. There is a large untapped potential for optimization, model ranking and uncertainty quantification. Here, we scratch the surface of what will likely be a more important topic in the coming years. Multiscale solvers subdivide the domain, and consequently the problem, into smaller parts during the basis construction. The basis functions depend only on the local variations in the permeability and mobility field, which has previously been exploited to only update basis functions adaptively where the mobility has changed. In this work, we first consider problems where certain parameters that are normally static for a given simulation are considered to be uncertain. In particular, we consider the problem of fault multipliers, where there exist a significant uncertainty to how the fault impacts the flow field. A fault can be completely open to flow, or it may block off flow completely, or anywhere in-between. We also consider the problem of optimizing well controls using a global basis method that is accurate and efficient for models with a moderate number of wells, applying a more sophisticated optimization algorithm (BFGS) than the line search used in Paper II.

We use the flow diagnostics-techniques to evaluate the relative sweep efficiency of the different fault combinations using a single-phase solver. Multiscale solvers were considered to be a good fit with the flow diagnostics concept as they produce inexpensive, approximate solutions with an error that is often smaller than the parameter uncertainty of the underlying problem. For the problems with fault multipliers, we used a static multiscale basis, replacing only the basis functions that have support extending close to the faults under consideration. We also adapt the grid to the near-well region and close to the faults, which is made possible by the flexible coarse grid definition introduced for the MsRSB-method.

The results indicated that the relative ranking of the different fault realizations produced by the multiscale method were somewhat in agreement with the results fine-scale solver, but that a number of iterations was required to differentiate between fine-scale features such as small faults that did not significantly impact the flow field. In part, this was due to the sensitivity of the Lorenz coefficient to small errors in the flux field. The use of multiscale methods for fast evaluation of different equiprobable models with a large degree of similarity holds great promise, but further research on the choice of the ranking coefficients is required. The optimization results were positive, however, further establishing that the use of flow diagnostics with single-phase flow can be an excellent proxy model for a fully-featured multiphase simulation.

Paper VII

MRST-AD – An Open-Source Framework for Rapid Prototyping and Evaluation of Reservoir Simulation Problems

Stein Krogstad, Knut–Andreas Lie, Olav Møyner, Halvor Møll Nilsen,
Xavier Raynaud and Bård Skaflestad

In proceedings of the 2015 SPE Reservoir Simulation Symposium,
Houston, Texas, USA
DOI: 10.2118/173317-MS

The importance of a solid research framework is sometimes underestimated. The Matlab Reservoir Simulation Toolbox contains a large number of open source solvers and functions for solving different partial differential equations on unstructured grids. More recently, the use of automatic differentiation has enabled the toolbox to become a modeling tool for different physical effects as well, making it is straightforward to prototype new solvers with novel physical phenomena.

This paper summarizes the wide-ranging collective efforts towards bringing MRST into a more general prototyping platform with automatic differentiation as the backbone. One significant part of this effort is the object oriented framework (see Chapter 5) developed as a part of the doctoral dissertation that allows efficient reuse of different components, and drop-in replacement of almost any part of the simulator, whether the new component is a new linear solver, a time-stepping algorithm, a nonlinear solver or a set of new constitutive relationships. Software development is sometimes an under-appreciated part of research, but in the author's opinion, the quality of the work contained in this thesis would be greatly diminished without this framework to rapidly prototype and validate new, advanced solvers. Further demonstration of the flexibility of the framework is given in Paper IX. The papers that specifically deal with highly complex nonlinear problems (VII, IX, XIII and XIV) are all implemented in MRST AD-OO.

Releasing well-documented code as open source enables researchers to build upon the work of others. It is also an important step towards reproducible science, where code and data can be released alongside conventional routes for research dissemination like journal articles, talks or books.

Paper VIII

A Multiscale Restriction-Smoothed Basis Method for Compressible Black-Oil Models

Olav Møyner and Knut-Andreas Lie

SPE Journal, published ahead of print, June 2016
DOI: 10.2118/173265-PA

After the MsRSB-method had been validated for a wide variety of single- and two-phase test cases for incompressible flow, the natural next step was to consider compressible flow with gas dissolution, which is the workhorse of industry grade simulation. The properties and the gas dissolution present in the black-oil equations (2.16) have a strong dependency upon the pressure, which means that any multiscale method that aims to resolve the flow field is put to the test. For incompressible and immiscible physics, which are the most commonly considered model problem for published work on multiscale methods, there is no dependence on the pressure outside of the gradient used in Darcy's law. Because of this weak coupling, the pressure can be discontinuous, or off by orders-of-magnitude, as long as the velocity field is conservative: $\nabla \cdot \vec{v} = q$.

Although there exists a few published works for compressible and black-oil flow for different multiscale solvers (see e.g., [63, 52, 35]), this paper, to the authors' knowledge, represents the first paper where a multiscale method is applied to industry complexity grids and fluid physics. The numerical examples in this paper include the full SPE 10 model 2 scenario with no simplification to fluids; different compressibility studies on a modified version of the Norne field model [41], gas injection with the SPE 1 [93] fluid model, as well as a realistic, synthetic field model made by researchers at Heriot-Watt [8]. Altogether, the different test cases represent a through validation of the MsRSB applied to black-oil problems. We demonstrate the robustness of the method, and the possibility of systematically trading computational effort for increased accuracy. The solution strategy for these equations with the multiscale solver is shown in Figure 6.1 as supplementary to the paper.

Further work from this paper would include extension to related systems of equations with nonlinear effects, such as enhanced oil-recovery (see Paper IX), thermal problems [116] or CO_2 sequestration. The efficient pressure solver should also be complimented with a more sophisticated transport solver that exploits the physical properties of the transport problem, for instance by using higher-order discretizations, nonlinear reordering based on the flux graph [7, 58, 84], trust-region [46, 121, 65], adaptive gridding [31] or continuation Newton-type solvers [125, 103].

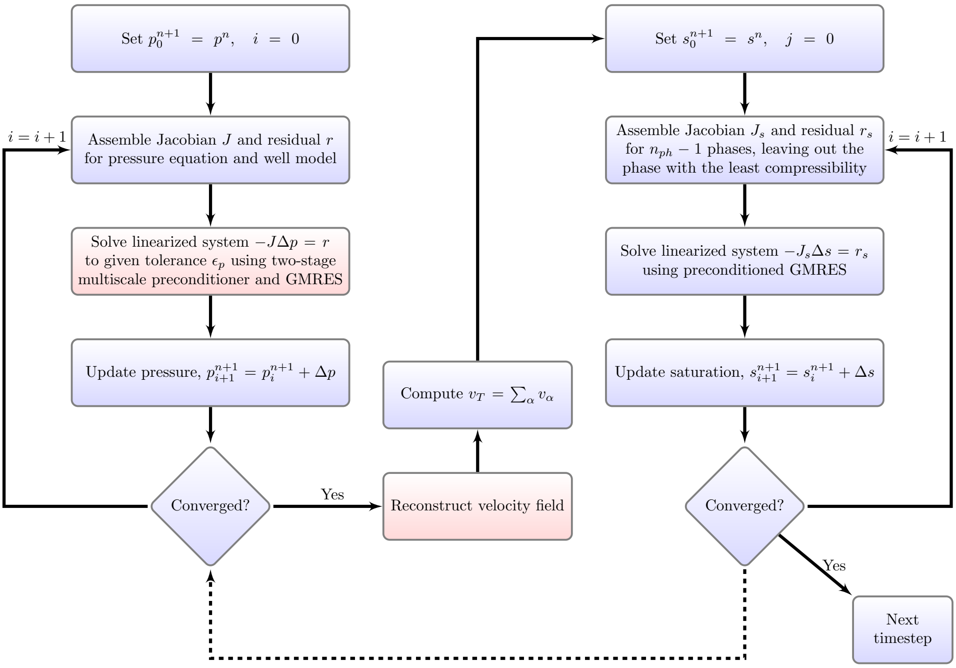

Set $p_0^{n+1} = p^n, \quad i = 0$
Assemble Jacobian J and residual r for pressure equation and well model
$i = i + 1$
Solve linearized system $-J\Delta p = r$ to given tolerance ϵ_p using two-stage multiscale preconditioner and GMRES
Update pressure, $p_{i+1}^{n+1} = p_i^{n+1} + \Delta p$
Converged?
Yes
Reconstruct velocity field
Compute $v_T = \sum_\alpha v_\alpha$
Set $s_0^{n+1} = s^n, \quad j = 0$
Assemble Jacobian J_s and residual r_s for $n_{ph} - 1$ phases, leaving out the phase with the least compressibility
$i = i + 1$
Solve linearized system $-J_s\Delta s = r_s$ using preconditioned GMRES
Update saturation, $s_{i+1}^{n+1} = s_i^{n+1} + \Delta s$
Converged?
Yes
Next timestep

Figure 6.1: Flowchart demonstrating the sequential implicit black-oil scheme used in Paper VIII. Note that the light blue boxes correspond to the sequential implicit scheme itself, while the light red boxes are steps that are performed only when the multiscale solver is used.

Paper IX

Multiscale Simulation of Polymer Flooding with Shear Effects

Sindre Hilden, Olav Møyner, Knut-Andreas Lie and Kai Bao

Transport in Porous Media, volume 113, issue 1, pp. 111–135, 2016
DOI: 10.1007/s11242-016-0682-2

The black-oil equations are considered sufficient for a wide range of primary and secondary oil recovery scenarios. For tertiary production where enhanced oil-recovery is employed, the equations will be modified with additional components and constitutive relationships. Such modifications include the addition of surfactant, different types of polymer, nutrients for increased microbacterial activity in the reservoir and solvents such as CO_2 or N_2. From a mathematical perspective, these extensions can drastically alter the nature of the underlying equations and introduce significant nonlinearities where rapid changes in mobility and solubility is present.

In this paper, we consider the extension of the MsRSB black-oil solver described in Paper VII to polymer flooding. Polymers are added to the injected water to increase the viscosity and consequently produce a more stable displacement. The water viscosity, normally taken to be almost constant, will now depend strongly on the polymer concentration. In addition, the polymer model considered includes shear-dependent viscosity, where the non-Newtonian polymer-water mixture changes viscosity depending on the magnitude of the fluid velocity. As the fluid velocity is predicted by the pressure equation, this makes the pressure system significantly more nonlinear and the fully-implicit commercial simulator used for comparison purposes used greatly reduced time-steps to converge.

The MsRSB-method, as previously observed for the black-oil model, gave excellent agreement with the fine-scale solvers, both fully-implicit and sequential implicit. The observed speed-up by using a multiscale method was significant, further validating the multiscale strategy for relevant fluid physics discretized on challenging grids. On the implementation side, note that the multiscale solver itself was not at all modified from the black-oil paper. Due to the highly modular structure of the AD-OO framework, the multiscale solver can automatically be applied for any sequential scheme implemented in same framework. Development of the sequential-implicit scheme for this model is challenging, as the polymer rheology makes the pressure equation more nonlinear, where errors in the velocity field will impact the accuracy of the viscosity in the polymer front. A natural extension is to consider different models for polymer, or EOR in general with the multiscale solver. Here, as in the black-oil case, there is an untapped potential for faster and more robust transport solvers.

Paper X

The Multiscale Restriction Smoothed Basis Method for Fractured Porous Media (F-MsRSB)

Swej Shah, Olav Møyner, Matei Tene, Knut-Andreas Lie and Hadi Hajibeygi

Journal of Computational Physics, volume 318, pp. 36–57, 2016
DOI: 10.1016/j.jcp.2016.05.001

Most porous media in the subsurface contain fractures at different scales to some degree. For many models, it may be sufficient to upscale the fractures into the permeability to capture the flow pattern, but for applications where the majority of fluid transport takes place via fractures, it is required to include the fractures in the model. Capturing the effects of fractures is especially important in the production of geothermal energy and in the recovery unconventional petroleum resources found in impermeable shales.

The inclusion of fractures in the simulation of porous media has been considered using a wide range of different techniques. One option is to explicitly discretize the fractures using small cells with the same width as the fracture aperture, but this will often lead to a large number of cells for large models with many fractures. Some authors consider the fractures as lower dimensional objects since they can be extremely thin, placing the fracture cells on the faces between cells (dual-porosity, dual-permeability models) or as discrete objects as in the discrete fracture model (DFM) [48, 49]. The embedded fracture model [62] is another model allows for general fracture geometries without grid modification. The larger fractures are treated as separate objects connected to the cells they intersect with by fracture-matrix conductivity indices that are computed analogously to the treatment of well indices. Smaller fractures at shorter length scales than the matrix blocks are homogenized into the permeability.

There have been previous work on fracture flow using multiscale solvers, for example [86, 100, 34]. In this paper we build upon the embedded fracture approach and extend the MsRSB-method to fractured porous media. An algorithm for growing support regions for basis functions in the fractures is developed, resulting in a robust multiscale fracture solver with automated coarsening for complex fracture networks. The results indicate that the MsRSB, given a moderate number of fracture degrees-of-freedom, can accurately capture the impact of fractures on flow discretized with the embedded fracture model.

Paper XI

Fully Implicit Simulation of Polymer Flooding with MRST

Kai Bao, Knut-Andreas Lie, Olav Møyner and Ming Liu

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands

DOI: 10.3997/2214-4609.201601880

This paper demonstrates the use of MRST as a prototyping platform, with a special emphasis on polymer flooding. The simulation of polymer injection is of general interest as it is one of the more widely deployed EOR-techniques. We demonstrate the importance of including the effects of polymer in the prediction of tertiary recovery scenarios. We also use the polymer equations as an example of a model that is a close relative of the black-oil equations, where researchers and practitioners may achieve rapid progress by extending already implemented models in MRST.

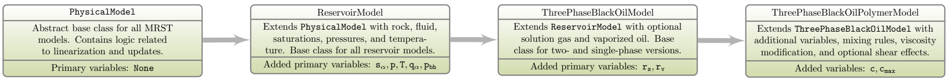

PhysicalModel
Abstract base class for all MRST models. Contains logic related to linearization and updates.
Primary variables: None
ReservoirModel
Extends PhysicalModel with rock, fluid, saturations, pressures, and temperature. Base class for all reservoir models.
Added primary variables: $s_\alpha, p, T, q_\alpha, p_{bh}$
ThreePhaseBlackOilModel
Extends ReservoirModel with optional solution gas and vaporized oil. Base class for two- and single-phase versions.
Added primary variables: r_s, r_v
ThreePhaseBlackOilPolymerModel
Extends ThreePhaseBlackOilModel with additional variables, mixing rules, viscosity modification, and optional shear effects.
Added variables: c, c_{max}

Figure 6.2: Model inheritance diagram for the polymer model, taken from Paper XI.

The object oriented framework is central to achieving this goal. Figure 6.2 (Figure 5 in the paper) demonstrates how the polymer model is a subclass of the black-oil model, which again is a subclass of the generic reservoir model, which again is one possible extension of the virtual physical model base class. As different parts of the framework are distinct, modular components, we automatically benefit from automatic time-stepping and advanced linear solvers for the new models that implement three-phase polymer flooding with shear dependent viscosity. The feature-set demonstrated in this paper is similar to the multiscale polymer simulator in Paper XI, and as such this paper compliments the aforementioned paper by demonstrating some of the inner workings of the infrastructure required to apply multiscale solvers in a generic manner.

Paper XII

Successful Application of Multiscale Methods in a Real Reservoir Simulator Environment

Knut-Andreas Lie, Olav Møyner, Jostein Roald Natvig, Antonina Kozlova, Kyrre Bratvedt, Shingo Watanabe and Zhouyi Li

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601893

Almost two decades after their introduction, multiscale solvers have yet not seen widespread commercial adoption. In part, this is due to the significant complexity of commercial simulators where the grids, physics and well models are often more complex than what is found in academic implementations. The research branch of the commercial INTERSECT reservoir simulator is developing a possible next-generation simulator with multiscale solvers as a key component. The research in this thesis is done as a part of an ongoing research collaboration to develop multiscale technology suitable for the tough requirements posed by a commercial grade code, including support for advanced physics, unstructured grids, parallelism and robust, automated coarsening.

MsRSB was introduced to the group of researchers and developers working on the INTERSECT multiscale project in September 2014 and before the end of the same year, it was implemented and used to simulate unstructured field models. MsRSB is currently the standard basis functions for the research simulator due to its ability to handle fully unstructured grids and superior robustness as well as accuracy over the classical MsFV basis functions. In this paper, we review the major developments in multiscale solvers for porous media over the previous two decades, before demonstrating the INTERSECT multiscale simulator on both a conceptual model with explicit fractures, as well as two field models. The Gullfaks field example that concludes the paper is a real black-oil field model without simplifications, simulated with a true multiscale reservoir simulator.

Paper XIII

Nonlinear Solver for Three-phase Transport Problems Based on Approximate Trust Regions

Olav Møyner

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601899

Implicit solvers for the transport equations (2.14) discretized by backward Euler are unconditionally stable. A sometimes under-emphasized point is that the unconditional stability does not necessarily mean that the solution of the resulting nonlinear equations is simple to find. As variants of Newton's method are normally used for the strongly nonlinear transport equations, the convergence to the right solution cannot be guaranteed if the residual function contains multiple Newton-contraction regions.

One approach for guaranteed convergent transport solvers was introduced by Jenny et al. [46] for S-shaped flux functions and later expanded by Wang and Tchelepi [121] and Li and Tchelepi [65] for flow with buoyancy and discretized fluxes, respectively. The approach, termed trust-region solver[1] uses strategic chopping of the solution increments produced by Newton's method in order to guarantee convergence.

Previous approaches to these types of solvers had promising results, but were restricted to two-phase problems where the solution variable is a single scalar per cell. As the chopping strategy relies on knowing inflection points and upwind changes a priori, these have to either be calculated manually or tabulated before simulation for the flow regimes that will be encountered. In this paper, a novel approach is presented that extends this idea to multicomponent flow with general relative permeabilities. The convergence issues for Newton's method applied to transport problems are examined, followed by the formulation of *flux search* algorithm for multicomponent transport. This algorithm is formulated by parametrization of the numerical flux functions along the solution path predicted by the Newton solver. From this parametrization, directional derivatives and discontinuities are estimated to identify safe updates. Both a global and a local approach to chopping is presented, before the method is systematically validated on a range of flow scenarios, ranging from gravity segregation to the simulation of a full field model. The solver is demonstrated to be capable of handling time-steps

[1] To avoid confusion, it should be noted that the trust regions in this context is different from trust regions for Newton's method applied to optimization problems.

of arbitrary length for highly nonlinear problems, including simulating 50 years of field prediction with strong gravity effects in a single time-step.

Robust transport solvers is important for a wide range of applications, as convergence failures and subsequent time-step cuts may significantly slow down simulations and reduce the ability to evaluate different scenarios and outcomes of engineering decisions. One natural extension of this work would be to consider CO_2 sequestration, where the long time horizons for safe storage of greenhouse gases contrasts with the relatively short time-scale of viscous flow with strong gravity segregation effects. Another possible extension would be to thermal and EOR-processes, where the strong nonlinearities due to additional local parameters can lead to severe time-step restrictions.

Paper XIV

A Multiscale Restriction-Smoothed Basis Method for Compositional Models

Olav Møyner and Hamdi Tchelepi

Accepted for the proceedings of the SPE Reservoir Simulation Conference 2017, Woodlands, Texas, USA

For the final paper of the thesis, we consider an extension of the MsRSB-method to compositional problems. This includes the development of both fully-implicit and sequential implicit compositional simulators for MRST with a generalized cubic equation-of-state. The compositional equations cover a wide range of different scenarios and allow simulation with any number of different molecular species present. As the computational cost of a fully-implicit multicomponent simulation can be excessive, a robust sequential scheme with a fast multiscale solver is especially attractive.

The presented paper is the first compositional multiscale finite-volume solver with a cubic equation of state. We use the sequential mass formulation previously considered for the black-oil equations and the classical MsFV-method in [35]. The results indicate that the MsRSB-method gives excellent pressure and flux approximations for the compositional pressure equation. We observe that the new compositional solver in MRST agrees well with the Stanford research simulator AD-GPRS for a simple one-dimensional validation problem.

Examples with both CO_2 and water injection above and below the minimum miscibility pressure indicate that the correct phase behavior is predicted by the sequential scheme compared to the fully-implicit. The multiscale solver agrees with the implicit solver, although minor errors in the saturation profile is observed due to the non-converged pressure. For the final example, we demonstrate that a N_2 injection scenario on the Norne field model was accurately reproduced by the multiscale solver, even though some oscillations are observed in the producer well curves after gas breakthrough for the sequential scheme. Although this does not appear to be a stability issue in the conventional IMPES sense, improving the sequential scheme should be a priority for further work in order to efficiently apply multiscale solvers.

Chapter 7

Conclusion and future work

7.1 Concluding remarks

This thesis is the culmination of little over three years of research into understanding and improving the simulation of flow and transport in porous media. At the outset, the primary goal was to extend the MsFV-method to unstructured grids, a goal which was in the author's opinion met by introducing the MsRSB-method in Paper V. In a sense, this paper is the core of the thesis as much of the later research has been focused on extending the MsRSB-method to different problems with more sophisticated types of physics.

The work on flow diagnostics is complimentary to the work on multiscale methods, as they demonstrate the value of information contained in the flow fields that the multiscale methods strive to capture. The presented trust region transport solver considers the transport problem, which is very different from the pressure equation: The idealized transport problem is hyperbolic and local, inexpensive to solve and nonlinear where the idealized pressure problem is elliptic and global, expensive to solve and linear, demonstrating how all aspects of a simulation must be improved if one hopes to achieve improvement for the entire problem. Finally, the work on a flexible prototyping framework for flow and transport in porous media has been essential to achieve the results in this thesis, as the time spent developing good tools have given savings of a much greater magnitude when working with advanced fluid physics.

7.2 Future work

There are several possible avenues for extending the work in this thesis in new directions. For many of the choices made, we could have taken other avenues and obtained different results. We will consider the possibilities grouped according to the same categorization of themes as in Chapter 6.

- For the development of multiscale methods for unstructured grids, there is a largely unexplored potential for considering different coarsening algorithms. In this thesis, we have considered the use of Metis for automated coarsening, which works reasonably well with MsRSB, but it seems likely that this result can be improved upon with coarsening that for example adapts to the flow field, analogous as the work done for transport problems by Hauge et al. [37]. We did not consider enrichment strategies for adaptive improvement of the solution, which is another field with active research [25, 77]. Finally, the tracer partitioning used to formulate the MsTPFA-method may have potential for the construction of tailored multiscale basis functions near specific features such as wells or faults where a local flow problem can determine strong connections.

- The use of flow diagnostics with multiscale solvers is a combination that holds great promise for different applications with uncertainty, including optimization problems and model ranking. One possible direction would be to use the smoothing iterations of MsRSB to capture minor changes to the permeability field for similar geostatistical realizations in order to rapidly evaluate large ensembles with flow diagnostics-techniques.

- In terms of building multiscale solvers for compressible and compositional flow problems, there is still a large number of different models that have not yet been considered. Preliminary work on applying the MsRSB-method to thermal problems is discussed by Vennemo [116]. While there are many different EOR-models that could be possible extensions, thermal problems are especially attractive as the temperature can strongly influence the coefficients of any pressure equation. More robust sequential implicit schemes should also be a priority, especially schemes that have rigorous estimates for when the pressure equation should be re-visited after transport to obtain the correct solution. Another natural extension would also be to use higher order schemes for the transport equations in order to better resolve the fronts of the displacement, which may be more feasible when the pressure is inexpensive to solve due to the (approximate) multiscale solver.

- The trust-region solver for multiphase flow can easily be extended to more components. EOR problems are a natural fit, as they extend the existing transport model with additional components that impact the fractional flow curves. Thermal problems are also interesting, as they appear to have convergence issues even for fully-implicit problems that are not always well understood. The implementation described in Paper XIII uses first-order potential upwind together with Newton's method, but other discretization and solver strategies may also benefit from the trust-region approach. Examples include hybrid upwinding [61, 36], the C1-PPU scheme presented by Jiang and Younis [47], reordering methods [7, 58, 84] or higher-order schemes for transport.

- The object-oriented prototyping framework in MRST is built from the ground up to be extensible to other types of equations and the framework is used in the development of new simulators in the Computational Geosciences group at SINTEF and other places. Extensive testing has demonstrated that the framework makes it simple to extend and implement new solvers and models based on the existing capability for black-oil, compositional and EOR. New researchers can easily leverage the work of others to demonstrate new ideas and the open source license makes it easy to share not only the results, but also the code behind a research paper.

Bibliography

[1] J. E. Aarnes, S. Krogstad, and K.-A. Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, 5(2):337–363, 2006. ISSN 1540-3459. doi: 10.1137/050634566.

[2] J. E. Aarnes, S. Krogstad, and K.-A. Lie. Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.*, 12(3):297–315, 2008. ISSN 1420-0597. doi: 10.1007/s10596-007-9072-8.

[3] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Comput. Geosci.*, 6:405–432, 2002. doi: 10.1023/A:1021291114475.

[4] I. Aavatsmark, E. Reiso, and R. Teigland. Control-volume discretization method for quadrilateral grids with faults and local refinements. *Comput. Geosci.*, 5:1–23, 2001. doi: 10.1023/A:1011601700328.

[5] F. O. Alpak, M. Pal, and K.-A. Lie. A multiscale method for modeling flow in stratigraphically complex reservoirs. *SPE J.*, 17(4):1056–1070, 2012. doi: 10.2118/140403-PA.

[6] J. Appleyard. Nested factorization. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1983.

[7] J. R. Appleyard and I. M. Cheshire. The cascade method for accelerated convergence in implicit simulators. In *European Petroleum Conference*, pages 113–122, SPE 12804, 1982. doi: 10.2118/12804-MS.

[8] D. Arnold, V. Demyanov, D. Tatum, M. Christie, T. Rojas, S. Geiger, and P. Corbett. Hierarchical benchmark case study forhistory matching, uncertainty quantification and reservoir characterisation. *Computers & Geosciences*, 50:4–15, 2013. doi: 10.1016/j.cageo.2012.09.011.

[9] L. Baker. Three-phase relative permeability correlations. In *SPE Enhanced Oil Recovery Symposium*. Society of Petroleum Engineers, 1988.

[10] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53 (6):2498–2508, 2014. doi: 10.1021/ie4033999. URL `http://pubs.acs.org/doi/abs/10.1021/ie4033999`.

[11] I. Berre, K. H. Karlsen, K.-A. Lie, and J. R. Natvig. Fast computation of arrival times in heterogeneous media. *Computational Geosciences*, 9(4): 179–201, 2005.

[12] Y. Brenier and J. Jaffré. Upstream differencing for multiphase flow in reservoir simulation. *SIAM J. Numer. Anal.*, 28(3):685–696, 1991. doi: 10.1137/0728036.

[13] F. Brezzi, K. Lipnikov, and V. Simoncini. A family of mimetic finite difference methods on polygonial and polyhedral meshes. *Math. Models Methods Appl. Sci.*, 15:1533–1553, 2005. doi: 10.1142/S0218202505000832.

[14] R. H. Brooks and A. T. Corey. Hydraulic properties of porous media and their relation to drainage design. *Transactions of the ASAE*, 7(1):26–0028, 1964.

[15] G. L. Brown, D. A. Collins, Z. Chen, et al. Efficient preconditioning for algebraic multigrid and red-black ordering in adaptive-implicit black-oil simulations. In *SPE Reservoir Simulation Symposium.* Society of Petroleum Engineers, 2015.

[16] H. Cao. *Development of techniques for general purpose simulators.* PhD thesis, Stanford University Stanford, CA, 2002.

[17] H. Cao, H. A. Tchelepi, J. R. Wallis, and H. E. Yardumian. Parallel scalable unstructured cpr-type linear solver for reservoir simulation. In *SPE Annual Technical Conference and Exhibition.* Society of Petroleum Engineers, 2005.

[18] Y. Chen, L. J. Durlofsky, M. Gerritsen, and X.-H. Wen. A coupled local–global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, 26(10):1041–1060, 2003.

[19] M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, 4:308–317, 2001. doi: 10.2118/72469-PA. Url: `http://www.spe.org/csp/`.

[20] K. Coats. Impes stability: The cfl limit. In *SPE Reservoir Simulation Symposium.* Society of Petroleum Engineers, 2001.

[21] K. Coats. Impes stability: Selection of stable timesteps. *SPE Journal*, 8(02): 181–187, 2003.

[22] K. H. Coats. Simulation of Gas Condensate Reservoir Performance. *Journal of Petroleum Technology*, 37(10):1870–1886, 1985. ISSN 0149-2136. doi: 10.2118/10512-PA. URL `http://www.onepetro.org/doi/10.2118/10512-PA`.

[23] K. H. Coats. A note on impes and some impes-based simulation models. *SPE Journal*, 5(03):245–251, 2000.

[24] D. Collins, L. Nghiem, Y. Li, and J. Grabonstotter. An efficient approach to adaptive-implicit compositional simulation with an equation of state. *SPE reservoir engineering*, 7(02):259–264, 1992.

[25] D. Cortinovis and P. Jenny. Iterative galerkin-enriched multiscale finite-volume method. *Journal of Computational Physics*, 277:248–267, 2014.

[26] A. H. Dogru, L. S. Fung, U. Middya, T. M. Al-Shaalan, J. A. Pita, K. HemanthKumar, H. Su, J. C. Tan, H. Hoy, and W. Dreiman. A next-generation parallel reservoir simulator for giant reservoirs. In *SPE/EAGE Reservoir Characterization & Simulation Conference*, 2009.

[27] M. G. Edwards and C. F. Rogers. A flux continuous scheme for the full tensor pressure equation. *Proc. of the 4th European Conf. on the Mathematics of Oil Recovery*, 1994.

[28] R. E. Ewing, T. F. Russell, and M. F. Wheeler. Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics. *Computer Methods in Applied Mechanics and Engineering*, 47(1):73–92, 1984.

[29] P. Forsyth and P. Sammon. Practical considerations for adaptive implicit methods in reservoir simulation. *Journal of Computational Physics*, 62(2):265–281, 1986.

[30] T.-P. Fries and T. Belytschko. The extended/generalized finite element method: an overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010.

[31] R. H. J. Gmelig Meyling. A characteristic finite element method for solving non-linear convection-diffusion equations on locally refined grids. In D. Guerillot and O. Guillon, editors, *2nd European Conference on the Mathematics of Oil Recovery*, pages 255–262, Arles, France, Sept 11-14 1990. Editions Technip.

[32] S. Gries, K. Stüben, G. L. Brown, D. Chen, D. A. Collins, et al. Preconditioning for fficiently applying algebraic multigrid in fully implicit reservoir simulations. *SPE Journal*, 19(04):726–736, 2014.

[33] H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *J. Comput. Phys*, 227(19):8604–8621, 2008. doi: 10.1016/j.jcp.2008.06.013.

[34] H. Hajibeygi, D. Karvounis, and P. Jenny. A hierarchical fracture model for the iterative multiscale finite volume method. *Journal of Computational Physics*, 230(24):8729–8743, 2011.

[35] H. Hajibeygi, H. A. Tchelepi, et al. Compositional multiscale finite-volume formulation. *SPE Journal*, 19(02):316–326, 2014.

[36] F. P. Hamon and H. A. Tchelepi. Analysis of hybrid upwinding for fully-implicit simulation of three-phase flow with gravity. *SIAM Journal on Numerical Analysis*, 54:1682–1712, 2016.

[37] V. L. Hauge, K.-A. Lie, and J. R. Natvig. Flow-based coarsening for multiscale simulation of transport in porous media. *Computational Geosciences*, 16(2):391–408, 2012.

[38] Z. He, H. Parikh, A. Datta-Gupta, J. Perez, and T. Pham. Identifying reservoir compartmentalization and flow barriers from primary production using streamline diffusive time of flight. *SPE J.*, 7(3):238–247, June 2004. doi: 10.2118/88802-PA.

[39] T. Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[40] E. Idrobo, M. Choudhary, and A. Datta-Gupta. Swept volume calculations and ranking of geostatistical reservoir models using streamline simulation. In *SPE/AAPG Western Regional Meeting*, Long Beach, California, USA, 19–23 June 2000. SPE 62557.

[41] IO Center, NTNU. The Norne benchmark case. url: `http://www.ipt.ntnu.no/~norne/wiki/doku.php`, 2012.

[42] O. Izgec, M. Sayarpour, and G. M. Shook. Maximizing volumetric sweep efficiency in waterfloods with hydrocarbon f-ϕ curves. *Journal of Petroleum Science and Engineering*, 78(1):54–64, 2011. doi: 10.1016/j.petrol.2011.05.003.

[43] J. Jansen. Adjoint-based optimization of multi-phase flow through porous media–a review. *Computers & Fluids*, 46(1):40–51, 2011.

[44] P. Jenny, S. H. Lee, and H. A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.*, 187: 47–67, 2003. doi: 10.1016/S0021-9991(03)00075-5.

[45] P. Jenny, S. H. Lee, and H. A. Tchelepi. Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media. *J. Comput. Phys.*, 217(2):627–641, 2006. doi: 10.1016/j.jcp.2006.01.028.

[46] P. Jenny, H. A. Tchelepi, and S. H. Lee. Unconditionally convergent nonlinear solver for hyperbolic conservation laws with s-shaped flux functions. *Journal of Computational Physics*, 228(20):7497–7512, 2009.

[47] J. Jiang and R. Younis. C1-ppu schemes for efficient simulation of coupled flow and transport with gravity. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[48] M. Karimi-Fard, A. Firoozabadi, et al. Numerical simulation of water injection in 2d fractured media using discrete-fracture model. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers, 2001.

[49] M. Karimi-Fard, L. J. Durlofsky, K. Aziz, et al. An efficient discrete fracture model applicable for general purpose reservoir simulators. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2003.

[50] V. Kippe, J. E. Aarnes, and K.-A. Lie. A comparison of multiscale methods for elliptic problems in porous media flow. *Comput. Geosci.*, 12(3):377–398, 2008. ISSN 1420-0597. doi: 10.1007/s10596-007-9074-6.

[51] S. Klevtsov, N. Castelletto, J. White, and H. Tchelepi. Block-preconditioned krylov methods for coupled multiphase reservoir flow and geomechanics. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[52] S. Krogstad, K.-A. Lie, H. M. Nilsen, J. R. Natvig, B. Skaflestad, J. E. Aarnes, et al. A multiscale mixed finite element solver for three phase black oil flow. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009.

[53] S. Krogstad, K.-A. Lie, O. Møyner, H. M. Nilsen, X. Raynaud, and B. Skaflestad. MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In *SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas*, 2015. doi: 10.2118/173317-MS.

[54] S. Krogstad, K. Lie, H. Nilsen, C. Berg, and V. Kippe. Flow diagnostics for optimal polymer injection strategies. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[55] S. Krogstad, X. Raynaud, Nilsen, and H. Møll. Reservoir management optimization using well-specific upscaling and control switching. *Computational Geosciences*, 20(3):695–706, 2016.

[56] R. Künze and I. Lunati. An adaptive multiscale method for density-driven instabilities. *J. Comput. Phys.*, 231(17):5557–5570, 2012. doi: 10.1016/j.jcp.2012.02.025.

[57] Y. Kuznetsov and S. Repin. New mixed finite element method on polygonal and polyhedral meshes. *Russ. J. Numer. Anal. Math. Modellling*, 18(3):261–278, 2003.

[58] F. Kwok and H. Tchelepi. Potential-based reduced newton algorithm for nonlinear multiphase flow in porous media. *Journal of Computational Physics*, 227(1):706–727, 2007.

[59] S. Lacroix, Y. V. Vassilevski, and M. F. Wheeler. Decoupling preconditioners in the implicit parallel accurate reservoir simulator (ipars). *Numerical linear algebra with applications*, 8(8):537–549, 2001.

[60] C. Le Potier. Finite volume scheme satisfying maximum and minimum principles for anisotropic diffusion operators. *Finite volumes for complex applications V*, pages 103–118, 2008.

[61] S. Lee, Y. Efendiev, and H. Tchelepi. Hybrid upwind discretization of nonlinear two-phase flow with gravity. *Advances in Water Resources*, 82:27–38, 2015.

[62] S. H. Lee, M. Lough, and C. Jensen. Hierarchical modeling of flow in naturally fractured formations with multiple length scales. *Water Resources Research*, 37(3):443–455, 2001.

[63] S. H. Lee, C. Wolfsteiner, and H. Tchelepi. Multiscale finite-volume formulation for multiphase flow in porous media: Black oil formulation of compressible, three phase flow with gravity. *Comput. Geosci.*, 12(3):351–366, 2008. doi: 10.1007/s10596-007-9069-3.

[64] S. H. Lee, H. Zhou, and H. A. Tchelepi. Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations. *J. Comput. Phys.*, 228(24):9036–9058, 2009. doi: 10.1016/j.jcp.2009.09.009.

[65] B. Li and H. A. Tchelepi. Unconditionally convergent nonlinear solver for multiphase flow in porous media under viscous force, buoyancy, and capillarity. *Energy Procedia*, 59:404–411, 2014.

[66] K. Lie, S. Krogstad, I. S. Ligaarden, J. R. Natvig, H. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Comput. Geosci.*, 16:297–322, 2012. ISSN 1420-0597. doi: 10.1007/s10596-011-9244-4. URL `http://dx.doi.org/10.1007/s10596-011-9244-4`.

[67] K.-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST).* SINTEF ICT, `http://www.sintef.no/Projectweb/MRST/publications`, 2nd edition, December 2015.

[68] K.-T. Lim. A new approach for residual and Jacobian arrays construction in reservoir simulators. *SPE Computer Applications*, 7(04):93–96, 1995.

[69] K. Lipnikov, M. Shashkov, and I. Yotov. Local flux mimetic finite difference methods. *Numer. Math.*, 112(1):115–152, 2009. ISSN 0029-599X. doi: 10.1007/s00211-008-0203-5. URL `http://dx.doi.org/10.1007/s00211-008-0203-5`.

[70] J. Lohrenz, B. G. Bray, and C. R. Clark. Calculating viscosities of reservoir fluids from their compositions. *Journal of Petroleum Technology*, 16(10): 1–171, 1964.

[71] A. Lukyanov and C. Vuik. Parallel fully implicit smoothed particle hydrodynamics based multiscale method. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[72] I. Lunati and P. Jenny. A multiscale finite-volume method for three-phase flow influenced by gravity. In P. Binning, P. Engesgaard, H. Dahle, G. Pinder, and W. Gray, editors, *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, Copenhagen, Denmark, 18–22 June 2006. URL `http://proceedings.cmwr-xvi.org/`.

[73] I. Lunati and P. Jenny. Multiscale finite-volume method for compressible multiphase flow in porous media. *J. Comput. Phys.*, 216(2):616–636, 2006. doi: 10.1016/j.jcp.2006.01.001.

[74] I. Lunati and S. H. Lee. An operator formulation of the multiscale finite-volume method with correction function. *Multiscale Model. Simul.*, 8(1): 96–109, 2009. doi: 10.1137/080742117.

[75] I. Lunati, M. Tyagi, and S. H. Lee. An iterative multiscale finite volume algorithm converging to the exact solution. *J. Comput. Phys.*, 230(5):1849–1864, 2011. doi: 10.1016/j.jcp.2010.11.036.

[76] A. Manea, J. Sewall, and H. A. Tchelepi. Parallel multiscale linear solver for highly detailed reservoir models. In *SPE Reservoir Simulation Symposium.* Society of Petroleum Engineers, 2015.

[77] A. Manea, H. Hajibeygi, P. Vassilevski, and H. Tchelepi. Enriched algebraic multiscale linear solver. In *ECMOR XV-15th European Conference on the Mathematics of Oil Recovery*, 2016.

[78] J. J. Martin. Cubic equations of state - which? *Industrial & Engineering Chemistry Fundamentals*, 18(2):81–97, 1979.

[79] Metis website. Metis – serial graph partitioning and fill-reducing matrix ordering, 2012. url: `http://glaros.dtc.umn.edu/gkhome/views/metis`.

[80] M. L. Michelsen. The isothermal flash problem. part ii. phase-split calculation. *Fluid Phase Equilibria*, 9(1):21–40, 1982.

[81] M. L. Michelsen. The isothermal flash problem. part i. stability. *Fluid phase equilibria*, 9(1):1–19, 1982.

[82] O. Møyner and K.-A. Lie. A multiscale method based on restriction-smoothed basis functions suitable for general grids in high contrast media. In *SPE Reservoir Simulation Symposium held in Houston, Texas, USA, 23–25 February 2015*, 2015. doi: 10.2118/173256-MS. SPE 173265-MS.

[83] MRST. The MATLAB Reservoir Simulation Toolbox, 2015a, May 2012. `http://www.sintef.no/MRST/`.

[84] J. R. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements. *Journal of Computational Physics*, 227(24):10108–10124, 2008.

[85] J. R. Natvig, K.-A. Lie, B. Eikemo, and I. Berre. An efficient discontinuous galerkin method for advective transport in porous media. *Advances in water resources*, 30(12):2424–2438, 2007.

[86] J. R. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, and S. Khataniar. Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs. *SPE Journal*, 16(04):880–888, 2011.

[87] R. D. Neidinger. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM review*, 52(3):545–563, 2010.

[88] K. Nikitin, K. Terekhov, and Y. Vassilevski. A monotone nonlinear finite volume method for diffusion equations and multiphase flows. *Computational Geosciences*, 18(3-4):311–324, 2014.

[89] J. Nordbotten, I. Aavatsmark, and G. Eigestad. Monotonicity of control volume methods. *Numer. Math.*, 106(2):255–288, 2007. doi: 10.1007/s00211-006-0060-z.

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

[91] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic transactions on numerical analysis*, 37(6):123–146, 2010.

[92] ODbL. Open data commons open database license, May 2016. `http://opendatacommons.org/licenses/odbl/`.

[93] A. S. Odeh. Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. *Journal of Petroleum Technology*, 33(01):13–25, 1981.

[94] N. M. of Petroleum, Energy, and the Norwegian Petroleum Directorate. Norwegian petroleum facts, 2016. `http://www.norskpetroleum.no/en/facts-norwegian-oil-gas/`.

[95] D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *SPE Journal*, 23(03):531–543, 1983.

[96] D.-Y. Peng and D. B. Robinson. A new two-constant equation of state. *Industrial & Engineering Chemistry Fundamentals*, 15(1):59–64, 1976.

[97] M. Prévost, F. Lepage, L. J. Durlofsky, and J.-L. Mallet. Unstructured 3d gridding and upscaling for coarse modelling of geometrically complex reservoirs. *Petroleum Geoscience*, 11(4):339–345, 2005.

[98] O. Redlich and J. N. Kwong. On the thermodynamics of solutions. v. an equation of state. fugacities of gaseous solutions. *Chemical Reviews*, 44(1): 233–244, 1949.

[99] T. Russell et al. Stability analysis and switching criteria for adaptive implicit methods based on the cfl condition. In *SPE Symposium on Reservoir Simulation*. Society of Petroleum Engineers, 1989.

[100] T. H. Sandve, I. Berre, E. Keilegavlen, and J. M. Nordbotten. Multiscale simulation of flow and heat transport in fractured geothermal reservoirs: inexact solvers and improved transport upscaling. In *Thirty-Eighth Workshop on Geothermal Reservoir Engineering Stanford University, February*, pages 11–13, 2013.

[101] Schlumberger. *ECLIPSE 2013.2 Technical Description*, 2013.

[102] M. Shahvali, B. Mallison, K. Wei, and H. Gross. An alternative to streamlines for flow diagnostics on structured and unstructured grids. *SPE J.*, 17(3):768–778, 2012. doi: 10.2118/146446-PA.

[103] S. Sheth and R. Younis. Localized computation of newton updates in fully-implicit two-phase flow simulation. *Procedia Computer Science*, 80:1392–1403, 2016.

[104] G. Shook and K. Mitchell. A robust measure of heterogeneity for ranking earth models: The F-Phi curve and dynamic Lorenz coefficient. In *SPE Annual Technical Conference and Exhibition, 4-7 October, New Orleans, Louisiana*, 2009. doi: 10.2118/124625-MS.

[105] G. Soave. Equilibrium constants from a modified redlich-kwong equation of state. *Chemical Engineering Science*, 27(6):1197–1203, 1972.

[106] H. Stone. Probability model for estimating three-phase relative permeability. *Journal of Petroleum Technology*, 22(02):214–218, 1970.

[107] H. Stone. Estimation of three-phase relative permeability and residual oil data. *Journal of Canadian Petroleum Technology*, 12(4), 1973.

[108] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1):281–309, 2001.

[109] The Open Porous Media Initative. The Norne dataset. url: `https://github.com/OPM/opm-data`, 2015.

[110] G. Thomas, D. Thurnau, et al. Reservoir simulation using an adaptive implicit method. *Society of Petroleum Engineers Journal*, 23(05):759–768, 1983.

[111] J. A. Trangenstein and J. B. Bell. Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM J. Appl. Math.*, 49(3): 749–783, 1989.

[112] J. A. Trangenstein and J. B. Bell. Mathematical structure of compositional reservoir simulation. *SIAM J. Sci. and Stat. Comput.*, 10(5):817–845, 1989.

[113] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report 34, University of Colorado at Denver, Denver, CO, USA, 1994.

[114] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3): 179–196, 1996. doi: 10.1007/BF02238511.

[115] P. Vanek, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88(3):559–579, 2001. doi: 10.1007/s211-001-8015-y.

[116] S. B. Vennemo. Multiscale simulation of thermal flow in porous media. Master's thesis, The Norwegian University of Science and Technology (NTNU), 2016.

[117] P. Vinsome. Orthomin, an iterative method for solving sparse sets of simultaneous linear equations. In *SPE Symposium on Numerical Simulation of Reservoir Performance*. Society of Petroleum Engineers, 1976.

[118] D. V. Voskov and H. A. Tchelepi. Comparison of nonlinear formulations for two-phase multi-component eos based simulation. *Journal of Petroleum Science and Engineering*, 82:101–111, 2012.

[119] D. V. Voskov, H. A. Tchelepi, and R. Younis. General nonlinear solution strategies for multiphase multicomponent eos based simulation. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2009.

[120] J. R. Wallis. Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *SPE Reservoir Simulation Symposium, 15-18 November, San Francisco, California*, 1983. doi: 10.2118/12265-MS. SPE-12265-MS.

[121] X. Wang and H. A. Tchelepi. Trust-region based solver for nnonlinear transport in heterogeneous porous media. *Journal of Computational Physics*, 253: 114–137, 2013.

[122] Y. Wang, H. Hajibeygi, and H. A. Tchelepi. Algebraic multiscale solver for flow in heterogeneous porous media. *Journal of Computational Physics*, 259: 284–303, 2014.

[123] J. W. Watts. A compositional formulation of the pressure and saturation equations. *SPE J.*, 1(3):243–252, 1986.

[124] R. Younis, K. Aziz, et al. Parallel automatically differentiable data-types for next-generation simulator development. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2007.

[125] R. Younis, H. A. Tchelepi, and K. Aziz. Adaptively localized continuation-newton method–nonlinear solvers that converge all the time. *SPE Journal*, 15(02):526–544, 2010.

[126] H. Zhou and H. A. Tchelepi. Operator-based multiscale method for compressible flow. *SPE J.*, 13(2):267–273, 2008. doi: 10.2118/106254-PA.

[127] H. Zhou and H. A. Tchelepi. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. *SPE Journal*, 17(02):523–539, 2012.

[128] Y. Zhou, H. A. Tchelepi, and B. T. Mallison. Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2011.

[129] Y. Zhou, H. A. Tchelepi, and B. T. Mallison. Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2011.

[130] D. Zudkevitch and J. Joffe. Correlation and prediction of vapor-liquid equilibria with the redlich-kwong equation of state. *AIChE Journal*, 16(1):112–119, 1970.

Part II

Scientific Papers

Paper I

The Multiscale Finite Volume Method on Stratigraphic Grids

Olav Møyner and Knut-Andreas Lie

SPE Journal, volume 19, issue 5, pp. 816-831, 2014
DOI: 10.2118/163649-PA

Is not included due to copyright

Paper II

The Application of Flow Diagnostics for Reservoir Management

Olav Møyner, Stein Krogstad and Knut-Andreas Lie

SPE Journal, volume 20, issue 2, pp. 306–323, 2014
DOI: 10.2118/171557-PA

Is not included due to copyright

Paper III

A Multiscale Two-Point Flux-Approximation Method

Olav Møyner and Knut-Andreas Lie

Journal of Computational Physics, volume 275, pp. 273–293, 2014
DOI: 10.1016/j.jcp.2014.07.003

Journal of Computational Physics 275 (2014) 273–293

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A multiscale two-point flux-approximation method

Olav Møyner*, Knut-Andreas Lie

SINTEF ICT, Department of Applied Mathematics, P.O. Box 124, Blindern, N-0314 Oslo, Norway

ARTICLE INFO

Article history:
Received 23 October 2013
Received in revised form 6 June 2014
Accepted 7 July 2014
Available online 14 July 2014

Keywords:
Multiscale methods
Iterative multiscale methods
Unstructured grids
Prolongation operator
Reservoir simulation

ABSTRACT

A large number of multiscale finite-volume methods have been developed over the past decade to compute conservative approximations to multiphase flow problems in heterogeneous porous media. In particular, several iterative and algebraic multiscale frameworks that seek to reduce the fine-scale residual towards machine precision have been presented. Common for all such methods is that they rely on a compatible primal–dual coarse partition, which makes it challenging to extend them to stratigraphic and unstructured grids. Herein, we propose a general idea for how one can formulate multiscale finite-volume methods using only a primal coarse partition. To this end, we use two key ingredients that are computed numerically: (i) elementary functions that correspond to flow solutions used in transmissibility upscaling, and (ii) partition-of-unity functions used to combine elementary functions into basis functions. We exemplify the idea by deriving a multiscale two-point flux-approximation (MsTPFA) method, which is robust with regards to strong heterogeneities in the permeability field and can easily handle general grids with unstructured fine- and coarse-scale connections. The method can easily be adapted to arbitrary levels of coarsening, and can be used both as a standalone solver and as a preconditioner. Several numerical experiments are presented to demonstrate that the MsTPFA method can be used to solve elliptic pressure problems on a wide variety of geological models in a robust and efficient manner.

1. Introduction

The oil and gas industry has for decades been a big user of computing and data processing to plan cost effective exploration and production of petroleum resources. Advanced desktop workflow tools have removed many bottlenecks that previously existed between different application domains. However, runtime is still often a limitation on the use of reservoir simulation, despite the fact that the available computational power has been greatly increased in the last decades. In part, this is caused by a continued increase in the complexity of reservoir models, and in part by modern workflows such as model calibration, uncertainty quantification, and optimization of production and recovery, which all require fast, scalable, and accurate flow simulations, possibly for large ensembles of model realizations.

Multiscale simulation technology promises to improve runtime by at least an order of magnitude compared to current simulator performance for traditional reservoir engineering workflows and offers a systematic framework for increasing local model resolution. This makes multiscale simulation a better tool for characterizing volumetric sweep and locating bypassed and immobile oil compared with traditional upscaling approaches which always imply a loss of information when homogenizing fine-scale structures. While there exist a wide variety of multiscale methods tailored to an even wider range

* Corresponding author.
E-mail addresses: Olav.Moyner@sintef.no (O. Møyner), Knut-Andreas.Lie@sintef.no (K.-A. Lie).

http://dx.doi.org/10.1016/j.jcp.2014.07.003

of problems, the core idea of most multiscale methods developed for reservoir simulation is an attempt to separate local and global phenomena, giving fine-scale solutions based on coarse-scale problems with precomputed basis functions that account for localized features [1]. Multiscale formulations also provide a framework that enables variable-fidelity simulation that within orders of magnitude shorter simulation times can provide qualitatively correct predictions of flow patterns. Utilizing this, often less emphasized, aspect of multiscale methods may lead to new and innovative simulation-based workflows for building earth models, which cannot be achieved with current state-of-the-art reservoir simulators.

Although modeling approaches used by the industry today are predominantly structured, they still 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. Providing multiscale simulation capabilities for general grids with unstructured connections may accelerate the general adoption of unstructured and refined grids as a means of improved representation of deviated well paths and complex geological features. Likewise, greater flexibility with respect to coarse grids is needed to develop automated coarsening strategies that perform well for complex models. If possible, coarse partitions should adapt to complex features, such as wells, barriers and channels in a way that ensures optimal accuracy for a chosen level of coarsening.

This paper will focus on multiscale methods for computing pressure and fluxes on models that are realistic with the respect to geometrical description and petrophysical heterogeneity. Our main goal is to develop multiscale methods that are more accurate and simpler to use than state-of-the-art upscaling methods across a wide range of upscaling factors and at least an order-of-magnitude faster than conventional methods when used as a fine-grid solver. To provide value for commercial applications, the methods should work with industry-standard grid formats, be easy to implement within next-generation simulators, and generally be accurate, efficient, intuitive to use, and robust with respect to flow models and parameter choices. A lot of the necessary technological components have been developed over the last decade. The current industry-relevant state-of-the-art mainly exists as complementary technologies that are developed from two different mathematical principles: the multiscale finite-volume (MsFV) method [2] and the multiscale mixed finite-element (MsMFE) method [3,4].

The original MsFV method [2] was developed for solving flow problems with many scales and a predominantly elliptic nature, and used basis functions computed on a dual grid to define transmissibilities in a multi-point coarse-scale discretization combined with another set of flow problems on the primal grid to reconstruct conservative fine-scale fluxes. The method has later been extended with a large body of research, including adaptivity [5], correction functions [6–8], iterative variants for error control [9,10], and operator formulations suitable for pre-existing solvers [11,12,8], and has been applied to a wide range of physical problems, including compressible black-oil models [13] and compositional flow [14], all on Cartesian grids. Extending the MsFV method to realistic grids is generally encumbered by the requirement of a dual coarse partition. We have previously shown that it is possible to generate compatible primal–dual partitions for grids with realistic features like pinch-outs, erosions, faults and unstructured connections using a geometrical algorithm with topological post-processing [15,16]. However, the coarsening process is difficult to automate in a robust manner, puts restrictions on the coarsening degree and type of partitions that can be used, and will sometimes only give partition that drastically reduce solution quality. We have also encountered several real models (e.g., the Gullfaks model discussed in Section 3.4) on which we have not been able to construct a dual partition at all. In addition, it is well known that highly contrasted media and large anisotropy ratios pose problems for the reduced boundary condition used to construct the MsFV basis functions [17–20].

The ability to handle fully unstructured grids with almost arbitrary connected coarse-scale partitions is one of the main advantages of the MsMFE method. 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 basis functions in the MsMFE method are constructed by prescribing a unit flow across faces in the coarse grid, which ensures that the latter is more robust with respect to the size and shape of the coarse blocks when applied to stratigraphic and unstructured grids [21,22]. Pathological cases will of course also exist, but the accuracy of the MsMFE method can often be significantly improved by introducing a certain degree of adaption to local structures, particularly for high-contrast media [23,24]. For two-phase flow, the method has been shown to provide good accuracy on industry-standard geological models [25]. On the other hand, although iterative versions of the MsMFE method have been applied to compressible flow [26,27] and black-oil models [28,29], a full extension of the method to realistic flow physics is generally encumbered by the need for a robust splitting of flow and transport, in which the flow equations are written on mixed (hybrid) form.

In this paper, our aim is to overcome the complexity of constructing compatible primal–dual partitions *and* develop a formulation that gives more stable and robust coarse-scale systems on models with realistic geometries and heterogeneity. To this end, we present what we believe is a general approach that combines the best features of the two multiscale methods discussed above: multiscale finite-volume methods that use numerically constructed partition-of-unity functions to glue together elementary flow solutions associated with interfaces between coarse blocks. The partition-of-unity functions are crucial and will enable us to dispense with the requirement of a compatible primal–dual partition (also called a wire-basket ordering [30,31]) and hence provide the flexibility necessary to handle complex industry-standard grids with high aspect ratios and unstructured connections without significant impact to the solution quality. Likewise, using a finite-volume formulation enables the methods to efficiently resolve realistic flow physics, including capillarity, compressibility, and gravity, and, in particular, be applicable to industry-standard black-oil type models. The construction is fully algebraic, which means that existing multiscale techniques such as local stages and iterative cycles, followed by conservative fine-scale flux reconstructions [20], can be used without modification. Finally, using elementary flow solutions associated with interfaces instead of vertices means that the new multiscale methods will closely resemble traditional methods for transmissibility upscaling as well as more recent methods that focus on developing stable coarse-scale stencils [32–35].

Although our approach is general and can be applied to construct multiscale methods with multipoint coarse-scale connections, the details will only be developed for the case in which the interfaces consist of the entire face between two neighboring grid blocks. The result is a two-point flux-approximation method (called MsTPFA) in which each coarse-scale transmissibility captures the local properties of the differential operator in a localized region between the centers of two neighboring coarse blocks. For smooth heterogeneities, the resulting method may be less accurate than previous multiscale methods that feature multipoint coarse-scale connectivity. However, as our requirements for griding and petrophysical properties are quite demanding, we will be willing to sacrifice some accuracy if it leads to increased robustness and ease of implementation. The resulting MsTPFA method is easy to integrate in existing simulators either as a preconditioner or as a stand-alone solver. Moreover, although this is not elaborated and exploited herein, the method has a large degree of inherent parallelism that can likely be exploited to obtain an implementation with good scalability.

2. The multiscale two-point flux-approximation method

The starting point of our discussion is a discrete linear system,

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{q}, \tag{1}$$

which defines a fine-scale pressure problem. The system can be produced in any number of ways: from a stationary (single-phase) pressure equation, as a pressure subsystem extracted from a fully implicit system, or as a decoupled pressure equation in a sequential pressure–transport splitting,

$$-\nabla \cdot (\mathsf{K}\lambda_t \nabla p) = q, \tag{2}$$

for some total mobility λ_t and permeability K. Our numerical examples will be limited to the case when (2) is the discretization of an elliptic pressure equation. Although this is not a prerequisite for the formulation of the new multiscale method, we will in the following tacitly assume that the fine-scale discretization is based upon a two-point flux-approximation (TPFA). The TPFA method is the industry-standard discretization for stratigraphic grids, both corner-point and PEBI type, and while it may suffer from grid-orientation effects, the method is monotone.

The multiscale two-point flux approximation (MsTPFA) method is, like the name suggests, based on a coarse-scale operator that gives a two-point type stencil for computing global flow patterns on a coarse grid much like in traditional transmissibility upscaling. In addition, the method has a set of prolongation operators that allow reconstructions of conservative solutions on the underlying fine-scale grid, e.g., as in the MsFV and MsMFE methods. In the following, we will describe the construction of basis and correction functions, formulation of coarse-scale problem, and reconstruction of fine-scale solutions in more detail. Our main purpose is to develop a multiscale method that works well on the *discretized* cell-centered flow equations, which introduces several features that are not necessarily needed for the continuous flow equations. However, in an attempt to simplify the presentation, we will switch between a geometric and an algebraic (operator) descriptions of the method, trying to always use the description we believe is most transparent to the reader.

To motivate the method, let us first look a bit in detail on the MsFV method. When constructing coarse-scale systems, the MsFV method uses two coarse grids: The primal grid defines a typical coarse grid as used for upscaling, and the dual of this grid is then used to decompose the domain into local pressure problems. Once these problems have been solved and assembled into a basis, the interactions between the basis functions over the primal coarse grid gives a coarse system with multipoint connections. In addition, it is common to construct correction functions that take care of the particulate part of the solution that is not represented by the homogeneous basis functions. This is the geometrical description of the MsFV method. The method can also be described in algebraic form, using vector partitions and manipulation and elimination of matrix blocks in the fine-scale discretization [8,11,20]. This operator formulation is particularly useful when formulating the method on unstructured grids [15], but requires a compatible primal–dual partition (wire-basket ordering) that is generally difficult to construct for (industry-standard) grids with unstructured topologies. To avoid the complexity of constructing a wire-basket ordering, MsTPFA is designed using a single partition much in the same manner as in the MsMFE method and traditional transmissibility upscaling. MsTPFA still needs dual control volumes, but each control volume is associated with a given coarse face and can be constructed as a simple combination of fractions of the primal cells that are on opposite sides of the face.

To introduce our new method, we start by discussing the construction of basis functions for pressure, which consists of two parts. The first step is a preprocessing step that is performed once for a given partition to map local problems to the final basis. The second step is the solution of local pressure problems. We will describe the local pressure problems first, as they motivate the preprocessing step.

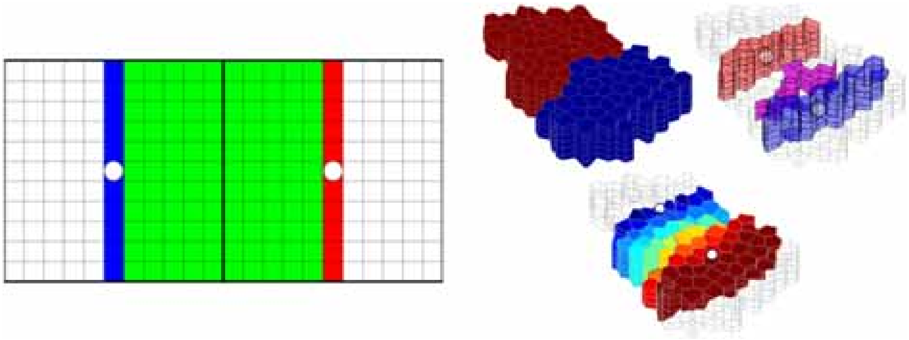


Fig. 1. Definition of local flow problems. The left plot shows two coarse blocks and a single face, with unit pressure specified to the left, zero pressure specified to the right, and pressure solution computed in the colored region in the middle. The three plots to the right show the same construction for a 2.5D PEBI grid: blocks sharing an interface, specification of fixed pressures, and local pressure solution, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.1. Local flow problems

Multiscale and most upscaling methods use local flow problems to estimate how fine-scale properties contribution to a coarse-scale equation that defines the global pressure field. The MsMFE method uses local problems defined over coarse faces and associates the degrees of freedom with the coarse faces. The MsFV method assigns local problems to the interaction regions around a coarse block, giving a single degree of freedom per coarse block. In the MsTPFA formulation, we solve local problems per coarse face, but we will finally associate the coarse system with a single degree of freedom per coarse block.

Fig. 1 illustrates how local flow problems are defined for each coarse face and restricted to the two coarse blocks that share the face. To further localize the problem, we introduce two additional boundaries that consist of all cells intersected by a plane through each block center defined by a normal vector directed towards the center point of the other block. In the left plot of Fig. 1, these cells are shown in color for each side of the coarse face, respectively. The local problem has unit pressure imposed on one side and zero on the other and no-flow boundary conditions are specified along the remaining outer boundaries of the local domain.

The local flow problem defined over a local domain Ω reads

$$-\nabla \cdot (\mathsf{K}\lambda_t \nabla p) = 0, \tag{3}$$

with boundary conditions

$$p = 1 \quad \text{on } \Gamma_{in}, \qquad p = 0 \quad \text{on } \Gamma_{out}, \qquad (\mathsf{K}\lambda_t \nabla p) \cdot \vec{n} = 0 \quad \text{on } \partial\Omega \setminus (\Gamma_{in} \cup \Gamma_{out}), \tag{4}$$

where Γ_{in} and Γ_{out} denote the inflow and outflow Dirichlet boundaries, respectively, and $\vec{n}$ denotes the normal vector. In the continuous formulation, this setup is identical to a popular variant of transmissibility upscaling, but in the discrete version it deviates slightly since we have chosen to impose the Dirichlet boundary as set of cells with prescribed pressure values rather than setting the Dirichlet boundary at a fixed location and discretizing the corresponding boundary conditions in a conventional manner. The motivation for doing so, is to simplify the algebraic construction of the method, which is the one we use in practice. To develop this construction from the existing global fine-scale equations, we extract all the rows corresponding to the cells in the local domain (i.e., the colored part of the left plot in Fig. 1) and permute the columns so that all the local unknowns appear in one block $\boldsymbol{p}_i$,

$$[\,\boldsymbol{A}_{i1} \quad \ldots \quad \boldsymbol{A}_{ii} \quad \ldots \quad \boldsymbol{A}_{in}\,] \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_i \\ \vdots \\ \boldsymbol{p}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1 \\ \vdots \\ \boldsymbol{q}_i \\ \vdots \\ \boldsymbol{q}_n \end{bmatrix}.$$

Then the local matrix is given by

$$(\boldsymbol{A}_{loc})_{km} = \begin{cases} (\boldsymbol{A}_{ii})_{km}, & k \neq m \\ (\boldsymbol{A}_{ii})_{kk} + \sum_{j\neq i} \sum_{\ell} (\boldsymbol{A}_{ij})_{k\ell}, & m = k \end{cases} \tag{5}$$

Further modifications are made to impose boundary conditions, i.e., replacing the entries of rows corresponding to the boundary with values one or zero. Linear interpolation is used when the centroid of a ‘Dirichlet boundary cell’ does not lie exactly on the plane that defines the boundary. Finally, the right-hand side of the local problem is set up by forcing correct values (i.e., zero everywhere except on the boundary). This is equivalent to solving a Dirichlet problem with constant pressure boundaries and no-flow on the boundary tangent to the flow direction.

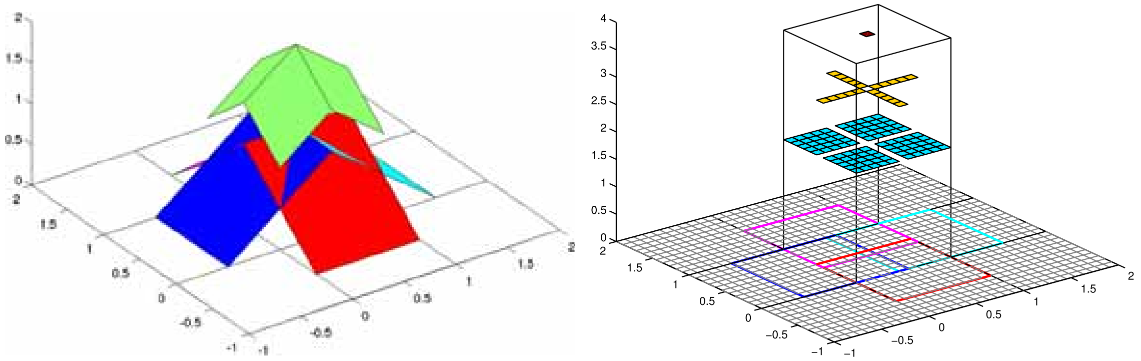


Fig. 2. The left plot shows the four elementary solutions p_ℓ ($\ell = E, W, N, S$) defined for a square block along with their superposition within the block. The right plot shows computational domains for the elementary solutions along with the sum, $\sum_\ell(\boldsymbol{p}_\ell + \hat{\boldsymbol{p}}_\ell)$, computed numerically inside the coarse block.

2.2. Partition of unity and definition of basis functions

It is possible to set up a conservative coarse-scale system based on the local flow solutions per face, much in the same manner as for transmissibility upscaling. However, obtaining a fine-scale pressure solution is more difficult. The local solutions can obviously not be used as basis functions directly. To illustrate this, we will look at a simple problem.

Example 1. Assume that the quadratic domain $[-1, 2] \times [-1, 2]$ is divided into quadratic blocks of unit size. For the block $[0, 1] \times [0, 1]$, we will have four different local solutions associated with the east, west, north, and south faces, respectively:

$$\begin{aligned} p_E(x, y) &= 1.5 - x, && (x, y) \in [0.5, 1.5] \times [0, 1], \\ p_W(x, y) &= 0.5 + x, && (x, y) \in [-0.5, 0.5] \times [0, 1], \\ p_N(x, y) &= 1.5 - y, && (x, y) \in [0, 1] \times [0.5, 1.5], \\ p_S(x, y) &= 0.5 + y, && (x, y) \in [0, 1] \times [-0.5, 0.5] \end{aligned}$$

that describe flow *out* over each face and into the neighboring block. The basis functions and their superposition restricted to the coarse center block are shown to the left in Fig. 2. If we look at the east face, for instance, there will be a similar elementary flow solution $\hat{p}_E = x - 0.5$ that describes flow from the neighbor and *into* the block. Obviously, these two elementary flow solutions must sum to unity, because if not, they would not be able to describe a constant pressure solution. Each pair of elementary solutions, $\boldsymbol{p}_\ell$ and $\hat{\boldsymbol{p}}_\ell$, computed with the discrete approach described above will by construction sum to unity inside their domain of definition. However, as illustrated in the right plot of Fig. 2, the local computational domains will have a non-trivial overlap, which means that it is not possible to represent a constant pressure using a simple superposition of the numerically computed elementary solutions, which is a requirement if such a superposition is to be useful as a multiscale finite-volume prolongation/basis operator.

To construct a useful basis, we will introduce an additional set of functions, one associated with each coarse face, that together will form a partition of unity. The purpose of these partition-of-unity functions is to extract the useful parts of the elementary solutions and discard the parts we are not interested in. That is, because the coarse-scale discretization seeks to mimic a two-point scheme we want to keep the part of the elementary functions that accurately represents the pressure drop between the two block centers on opposite sides of each face. Likewise, we wish to remove as much as possible of the local effects of the no-flow and Dirichlet boundary conditions used to localize the elementary solutions. Fig. 3 shows how such a partition of unity w_ℓ ($\ell = E, W, N, S$) can be constructed for the case discussed in Example 1. Here, the support of each function w_ℓ covers the corresponding coarse face and narrows in towards the two block centers to which it is associated. In general, if p^i_j denotes the elementary function defined between block i having unit pressure and block j having zero pressure and w^i_j denotes the corresponding partition-of-unity function, one can define a basis for each face,

$$\psi^i_j = w^i_j p^i_j \tag{6}$$

and a block basis that is simply the sum of the basis functions for all coarse faces in the block,

$$\psi^i = \sum_j \psi^i_j = \sum_j w^i_j p^i_j. \tag{7}$$

In Fig. 3, the middle plot shows the construction of face bases while the plot to the right shows the block basis. The basis function associated with the centroid of a coarse block has support on what would be considered the local neighborhood

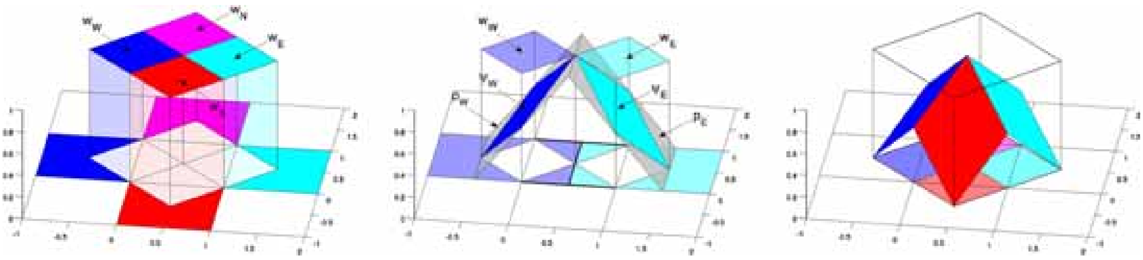


Fig. 3. Using partition of unity to define basis functions for the case in Example 1. The left plot shows the four functions w_ℓ ($\ell = E, W, N, S$) defining the partition of unity. The middle plot shows how partition functions w_E and w_W are used to extract parts of the elementary solutions p_E and p_W to define face bases ψ_E and ψ_W; thick lines indicate where boundary conditions (Dirichlet or no-flow) are specified to localize the problem in the coarse block. The right plot shows the resulting block basis with support on a rotated square.

for a coarse block using a TPFA approximation. (The MsFV method, on the other hand, has a support on what could be considered an MPFA-like neighborhood.)

The smoothness of the pressure approximation is inherited from the elementary solutions p^i_j and the partition functions w^i_j. In practice, one will obviously not use a discontinuous partition of unity as shown in Fig. 3. Indeed, if the partition functions are smooth and zero away from the boundaries used to localize the elementary functions, we will obtain an overall smooth pressure approximation.

Partition functions having the required properties can easily be constructed on almost arbitrarily shaped coarse block, regardless of whether the underlying grid is structured or unstructured. To motivate our approach, we consider a physical analogue. Assume that each coarse face is assigned a unique color and that ink with unit concentration of this color is injected uniformly along the face and a corresponding amount of fluid is extracted from point sources located at the block centers. If we continue injecting, a suitable partition of unity can be obtained from the steady-state ink concentrations inside each coarse block. Computationally, this is achieved by solving an incompressible pressure equation with homogeneous, isotropic permeability for each coarse block with a sink in the cell center and sources equal in magnitude to the face area on each face. Once the corresponding flow field $\vec{v}$ is obtained, the stationary tracer distribution for inflow boundary Γ can be computed from

$$\nabla \cdot (\vec{v}c) = 0, \quad c = 1 \text{ for } \vec{x} \in \Gamma. \tag{8}$$

The tracer equation (8) is then solved with a unique tracer for each coarse face, which gives values between zero and one for cells inside the coarse block. Fig. 4 illustrates the resulting computation of a face basis function for a homogeneous and a heterogeneous case.

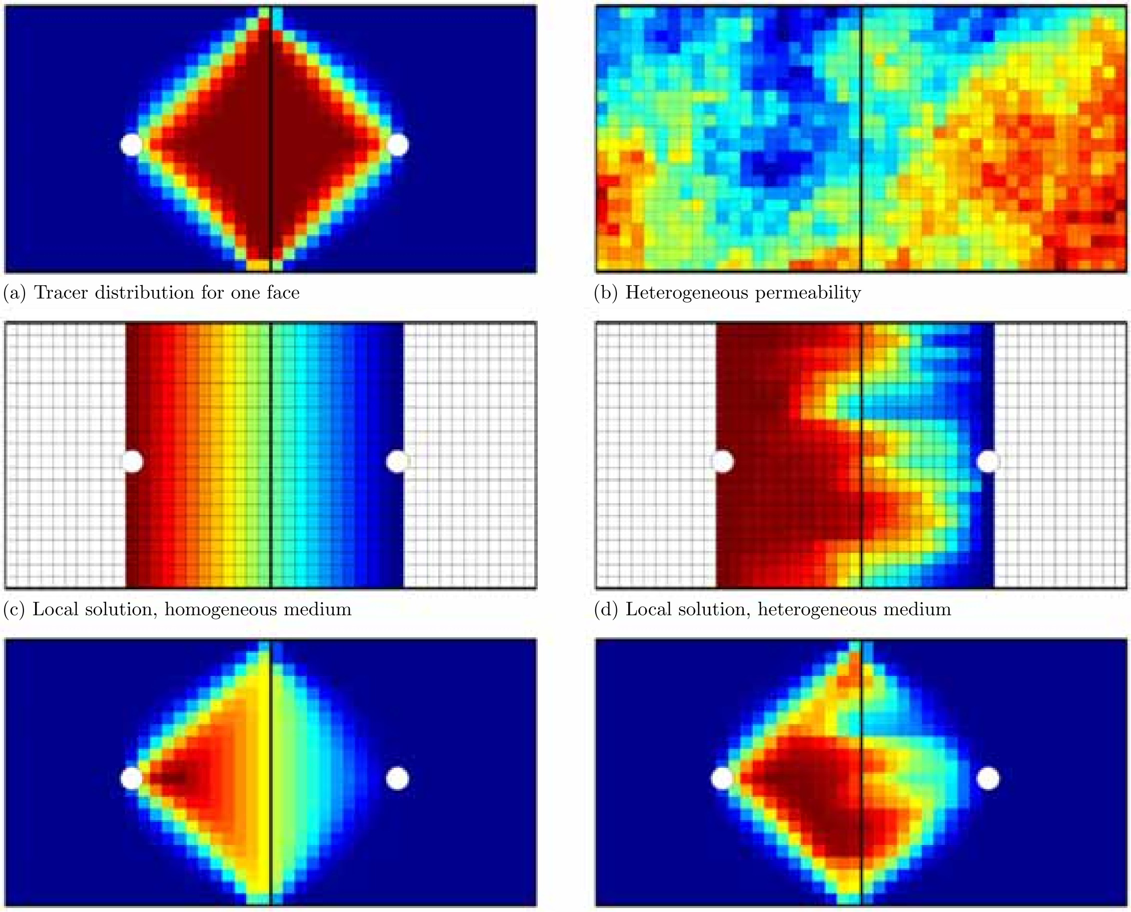
To solve the tracer equation, we use a standard first-order upwind discretization, which has an inherent numerical diffusion that will give us the required smoothness of the partition functions. If a two-point method is used to compute the flow field, the resulting fluxes will form a directed graph that can be used to reorder the discrete tracer equations into a triangular form that can be inverted efficiently by an algorithm that visits each cell once [36]. Hence, the cost of constructing partition functions is dominated by the cost of solving one pressure equation per block, an operation that is naturally parallel and only has to be performed once as the grid is partitioned into coarse blocks.

2.3. Correction functions and treatment of wells

While multiscale methods may produce a fine-scale pressure field from a coarse-scale problem, the basis functions only capture the elliptic fine-scale effects driven by a pressure differential. Other effects, e.g., body forces like gravity, will only be included in the coarse sense. In many cases, this may be sufficient to resolve the global pressure trends, but in other cases, one must also capture fine-scale effects that are not represented in the basis functions to properly resolve the global pressure trends. For an illustration of this we will consider a simple example.

Example 2. Let a rectangular discretized domain of $[0, 2] \times [0, 1]$ be divided into two unit-size blocks. Flow is driven by two injectors with $p_{bhp} = 1$ at $(1/4, 3/4)$ and $(3/4, 1/4)$ and a producer with $p_{bhp} = 0$ at $(3/4, 1/4)$, and no-flow boundary conditions ($\mathsf{K}\lambda_t \nabla p \cdot \vec{n} = 0$) are imposed on the boundary. The resulting solution is shown to the left in Fig. 5. If the coarse blocks are used along with regular basis functions to compute a fine-scale pressure, any combination of wells giving the same integral flux over the coarse boundaries will result in the same fine-scale solution as shown in the middle plot. However, by constructing local correction functions that capture the fine-scale well behavior as explained below, the solution becomes unique as shown in the right plot in Fig. 5.

The concept of correction functions was originally formulated for the MsFV method [6] and can be straightforwardly extended to the MsTPFA method to resolve effects from gravity, wells, etc. The central idea is to decompose the fine-scale pressure into two parts,


(a) Tracer distribution for one face

(b) Heterogeneous permeability

(c) Local solution, homogeneous medium

(d) Local solution, heterogeneous medium

(e) Face basis function, homogeneous medium

(f) Face basis function, heterogeneous medium

Fig. 4. Numerically computation of face basis for a homogeneous and a heterogeneous 2D medium.

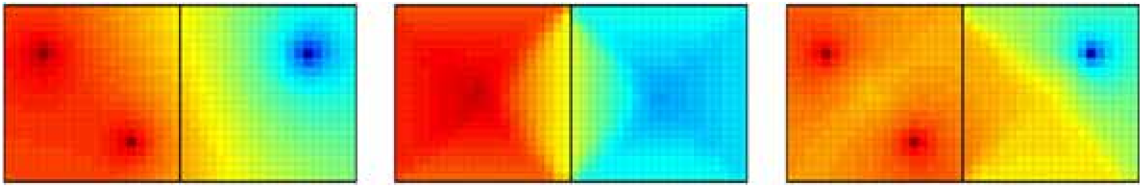


Fig. 5. The reference pressure solution (left) contains fine-scale information not present in the prolonged coarse multiscale solution (middle) which can be reintroduced by the use of correction functions (right).

$$\boldsymbol{p} = \boldsymbol{B}\boldsymbol{p}_c + \boldsymbol{c}, \tag{9}$$

where $\boldsymbol{c}$ is a *correction term* (or particular solution) that resolves fine-scale effects not accounted for by the *homogeneous* basis functions. Whereas the basis functions solve a flow problem (3) driven by a pressure differential (4), the correction functions are defined as solutions to an *inhomogeneous* problem driven by a source term

$$-\nabla \cdot (\mathsf{K}\lambda_t \nabla p_c) = q, \tag{10}$$

with zero boundary conditions

$$p_c = 0 \quad \text{on } \Gamma_i \cup \Gamma_o, \qquad (\mathsf{K}\lambda_t \nabla p_c) \cdot \vec{n} = 0 \quad \text{on } \partial\Omega \setminus (\Gamma_i \cup \Gamma_o). \tag{11}$$

Here, Γ_i and Γ_o denote the inflow and outflow boundaries of the corresponding homogeneous problem (4). The source term q can represent gravity, well models and other fluid sources and sinks, etc.

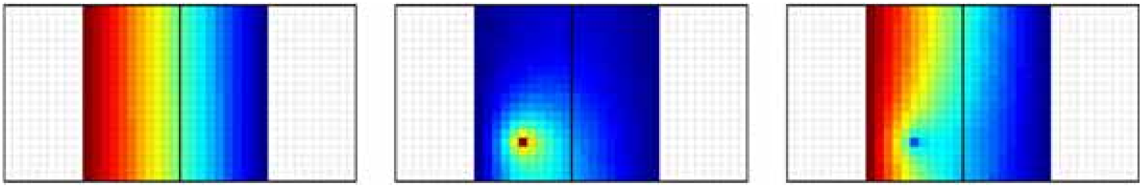


Fig. 6. The choice of local problem for a coarse face depends on the use of correction functions. Without correction functions, the local problem is defined by $\boldsymbol{A}_0$ and is completely linear (left). Formulated in companionship with a correction function (middle), the local basis problem accounts for the flux out of the well perforation and is constructed using $\boldsymbol{A}_0 + \boldsymbol{D}_{bc}$ (right).

Once the local correction problem (10)–(11) has been solved for each coarse face, the solutions can be combined with the partition of unity to define the correction operator on the whole domain in a similar manner as Eqs. (6) and (7), producing a local solution for each pair of coarse blocks (i, j)

$$\boldsymbol{c}_j^i = w_j^i p_c \tag{12}$$

Notice that since correction functions are constructed in the same manner as the elementary functions, they must be weighted by the partition of unity to avoid overlap effects in all cells that have more than one nonzero w_j^i. Moreover, because the correction term is independent of the coarse-scale problem, one can simply sum over all block-pairs to get the final correction over the entire domain

$$\boldsymbol{c} = \sum_{(i,j)} \boldsymbol{c}_j^i = \sum_j w_j^i p_c. \tag{13}$$

The fine-scale information present in correction functions representing wells and boundary conditions should replace and not add to the prolonged solution. It is therefore important that the flow problems (3) used to construct the corresponding homogeneous bases are modified locally to avoid interfering with the correction function. This is done by setting any pressure or flux equivalent to wells and boundary conditions to zero to leave 'gaps' in the solution of the prolonged system that are compatible with the correction functions. Mathematically, one replaces (3) with

$$-\nabla \cdot (\mathsf{K}\lambda_t \nabla p) = \tilde{q}, \tag{14}$$

where $\tilde{q}$ represents the flux from any boundary conditions or wells set to a zero value. In the operator formulation, this is done implicitly. Any cell attached to a boundary with prescribed condition different from no-flow or containing a well perforation will have extra diagonal entries in the linear system corresponding to the boundary transmissibility or well model used. Thus one can extract local linear systems and solve these with zero right hand side. That is, one splits the fine-scale system matrix $\boldsymbol{A}_{tot}$ into one part $\boldsymbol{A}_0$ representing cell-wise flux balances and a diagonal part $\boldsymbol{D}_{bc}$ representing flux contributions from boundary conditions and wells,

$$\boldsymbol{A}_{tot}\boldsymbol{p} = (\boldsymbol{A}_0 + \boldsymbol{D}_{bc})\boldsymbol{p} = q. \tag{15}$$

If the basis functions will be used alongside correction functions, we set $\boldsymbol{A} = \boldsymbol{A}_0 + \boldsymbol{D}_{bc}$ in (5). On the other hand, if the solution will be constructed without correction functions, it is natural to set $\boldsymbol{A} = \boldsymbol{A}_0$ and impose boundary conditions and wells at the coarse scale. In some cases, including inhomogeneous effects on the fine scale may not be required to obtain an acceptable approximation, while in other cases these effects can be accounted for by iterative techniques [20] or coarse-grid refinement. The effect of omitting inhomogeneous fine-scale effects is illustrated in Fig. 6, in which the basis function for a single coarse face from Example 2 is shown with and without correction functions.

2.4. Coarse system

The coarse-scale system for the MsTPFA method is straightforward to formulate, owing to the construction of local basis problems over coarse edges. Applying the divergence theorem to (2) for a coarse block Ω_i,

$$-\int_{\Omega_i} \nabla \cdot \mathsf{K}\lambda_t \nabla p = -\int_{\partial\Omega_i} (\mathsf{K}\lambda_t \nabla p) \cdot \vec{n} = \int_{\Omega_i} q. \tag{16}$$

Recall that the basis function for each coarse block is denoted ψ^i and let $\mathcal{N}(i)$ be the set of neighbors for coarse block i and $\tilde{p}_i$ be the coarse-scale pressure assigned to block i. This gives

$$\sum_{j\in\mathcal{N}(i)} T_{ij}(\tilde{p}_i - \tilde{p}_j) = -\int_{\partial\Omega_i} (\mathsf{K}\lambda_t \nabla\psi^i) \cdot \vec{n} = -\int_{\partial\Omega_i} \left(\mathsf{K}\lambda_t \nabla\left(\sum_j \psi_j^i\right)\right) \cdot \vec{n} = \int_{\Omega_i} q. \tag{17}$$

If we let Γ_{ij} denote the interface between blocks i and j, we can write $\partial\Omega_i = \bigcup_{j\in\mathcal{N}(i)} \Gamma_{ij}$. Moreover, because each local solution is defined to be nonzero for a single interface,

$$\int\limits_{\Gamma_{ik}} \left(\mathsf{K}\lambda_t \nabla \psi^i_j\right) \cdot \vec{n} = 0 \quad \text{for } k \neq j. \tag{18}$$

Hence, the transmissibility for each interface depends only on a single local solution, giving

$$T_{ij}(\tilde{p}_i - \tilde{p}_j) = -\int\limits_{\Gamma_{ij}} \left(\mathsf{K}\lambda_t \nabla \psi^i_j\right) \cdot \vec{n}, \qquad j \in \mathcal{N}(i). \tag{19}$$

Source terms are handled in the integral sense, with the optional correction functions c^i_j being subtracted to avoid accounting for fine-scale effects twice

$$Q_i = \int\limits_{\Omega_i} q + \sum_j \int\limits_{\partial\Omega_i} \left(\mathsf{K}\lambda_t \nabla c^i_j\right) \cdot \vec{n} \tag{20}$$

Once these quantities are computed, we can set up the coarse-scale problem

$$\boldsymbol{A}_c \boldsymbol{p}_c = \boldsymbol{q}_c \tag{21}$$

where the entries of the matrix and right hand side are

$$(\boldsymbol{A}_c)_{ij} = \begin{cases} T_{ij} & i \neq j \\ \sum_{j\in\mathcal{N}(i)} -T_{ij} & i = j \end{cases} \quad \text{and} \quad (\boldsymbol{q})_i = Q_i. \tag{22}$$

Finally, if we construct a prolongation operator $\boldsymbol{B}$ of size $N_f \times N_c$, in which entry i, j corresponds to the value of basis function (7) for coarse block j in fine cell i, the fine-scale pressure is approximated as

$$\boldsymbol{p} \approx \boldsymbol{B}\boldsymbol{p}_c + \boldsymbol{c}. \tag{23}$$

2.5. Conservative reconstruction

The multiscale fine-scale solution will not be conservative everywhere because, like most other multiscale methods, the initial pressure produced by the MsTPFA method only fulfills (1) approximately. If the resulting flux field is to be passed on to a fine-scale transport solver, it must be mass conservative on the fine scale to avoid unphysical values in the transport solver. For the MsFV method, this is typically handled by partially disconnecting the fine-scale flux field from the fine-scale pressure field, and solving local flow problems to reconstruct mass-conservative fluxes corresponding to the inexact multiscale pressure field. Herein, we will employ the same strategy as used in the MsFV method, which we for completeness will outline briefly below.

Conventional solvers let the fluxes be defined by the pressure, i.e., by applying Darcy's law directly,

$$\vec{v} = -K\lambda_t \nabla p. \tag{24}$$

As the coarse-scale pressure is resolved by a flux conservative system (19), the multiscale fine-scale pressure gives a conservative flux field over the coarse-block interfaces. We can exploit this by solving local flow problems

$$-\nabla \cdot (\mathsf{K}\lambda_t \nabla \bar{p}) = q, \tag{25}$$

in the local domain Ω with Neumann boundary conditions sampled from the block interfaces where the fine-scale pressure is conservative,

$$(\mathsf{K}\lambda_t \nabla \bar{p}) \cdot \vec{n} = (\mathsf{K}\lambda_t \nabla p) \cdot \vec{n} \quad \text{on } \partial\Omega. \tag{26}$$

We proceed to apply (24) to the reconstructed pressure $\bar{p}$, giving a new flux field that is conservative everywhere. For further discussion of these local problems as well as the operator analogue thereof the reader is referred to [11].

2.6. Iterative multiscale formulation

The pressure extrapolation described above gives an initial approximation to the fine-scale flow equations that may or may not be sufficiently accurate. To increase the accuracy of the multiscale approximation, the traditional MsFV method has been extended to an iterative formulation, the i-MsFV method [37,10], in which the mass-conservative multiscale operator is combined with an inexpensive iterative solver to systematically drive the fine-scale residual towards zero. In this multigrid-like approach, the coarse-scale multiscale operator will resolve global features in the pressure field, while the smoother is used to construct correction terms that account for errors in the prolongated pressure field.

If we let $\boldsymbol{p}_c^n$ denote the coarse pressure at step n and let $\boldsymbol{c}^n$ be some correction term defined such that

$$\left\| \boldsymbol{A}\left(\boldsymbol{B}\boldsymbol{p}_c^n + \boldsymbol{c}^n\right) - \boldsymbol{q} \right\| \le \left\| \boldsymbol{A}\boldsymbol{B}\boldsymbol{p}_c^n - \boldsymbol{q} \right\|, \tag{27}$$

a more accurate fine-scale pressure approximation is obtained by setting $\boldsymbol{p}_s^{n+1} = \boldsymbol{B}\boldsymbol{p}_c^n + \boldsymbol{c}^n$. While this pressure reduces the residual error of (1), it is no longer guaranteed to be conservative on the coarse scale. Hence, instead of using $\boldsymbol{p}_s^{n+1}$ directly, we will construct a new coarse-scale solution $\boldsymbol{p}_c^{n+1}$ that is mass-conservative and accounts for the residual effects of $\boldsymbol{c}$ so that

$$\boldsymbol{p}^{n+1} = \boldsymbol{B}\boldsymbol{p}_c^{n+1} + \boldsymbol{c}^n. \tag{28}$$

Inserting (28) into (1) and applying the discrete restriction operator

$$\chi_{ij} = \begin{cases} 1, & \text{if cell } j \text{ is in coarse block } i, \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

gives the coarse-scale problem

$$\chi \boldsymbol{A}\left(\boldsymbol{B}\boldsymbol{p}_c^{n+1} + \boldsymbol{c}_n\right) = \chi \boldsymbol{q}. \tag{30}$$

Defining $\boldsymbol{A}_c = \chi \boldsymbol{A}\boldsymbol{B}$ and moving the actions of the correction function to the right-hand side, the new coarse-scale systems reads,

$$\boldsymbol{A}_c \boldsymbol{p}_c^{n+1} = \chi\left(\boldsymbol{q} - \boldsymbol{A}\boldsymbol{c}^n\right). \tag{31}$$

Substituting the solution of (31) into (28) gives a new fine-scale pressure approximation that is conservative on the coarse scale since the coarse interface fluxes from $\boldsymbol{c}$ have been accounted for via the modified right-hand side. This is referred to as a *multiscale cycle*. It should be noted that writing $\boldsymbol{A}_c = \chi \boldsymbol{A}\boldsymbol{B}$ gives a coarse-scale discretization that is not strictly two-point because the basis functions may overlap at the vertices of the coarse blocks, which will result in additional weak multipoint couplings, i.e., couplings between coarse blocks that are not facial neighbors. To see this, think of a grid with a Cartesian ijk topology that has been given a tensor-product partition so that the coarse grid has a Cartesian IJK topology. Here, there will be weak couplings between block (I, J, K) and its neighbors $(I \pm 1, J \pm 1, K \pm 1)$, even if these blocks do not share a coarse face. Such couplings will not appear if the only connection between neighboring coarse blocks are through faces (e.g., as will be the case for 2D PEBI grids).

So far, no assumptions have been made about exactly *how* $\boldsymbol{c}$ is obtained, other than it should fulfill (27). In general, if we assume that the coarse-scale operator is able to resolve the global features of the system, any significant errors will be on the sub-scale. Hence, the local solver can be chosen in the same way as in traditional multigrid methods. A local solver for multiscale methods should thus be

- able to *reduce localized errors* present in the initial prolonged coarse-scale solution;
- *inexpensive* to construct and apply;
- *local* so that the domain-of-dependence for each node in the local solver should be small.

There are many methods that fulfill these requirements. For our purposes, the most important distinction is between solvers that require a setup phase and those who do not. Methods without a setup phase include iterative solvers such as Jacobi/Gauss–Seidel and block variants thereof, whereas methods with a setup phase are exemplified by various multigrid methods as well as LU-based preconditioners in which an incomplete factorization is used to approximate $\boldsymbol{A}$. Local solvers with a setup cost are likely useful when several multiscale cycles are required, while iterative solvers without setup cost are more useful when only a modest number of iterations are required, for instance when a single multiscale cycle gives satisfactory results. The choice of local solver and its impact on the convergence rate and computational cost of the overall multiscale method is thoroughly discussed by Wang et al. [20]. In particular, the authors show that correction functions are not very good local solvers. We have conducted an independent series of numerical experiments, which confirm these observations. Herein, however, our main focus is on the coarse-scale solver, and to illustrate the concept we will only utilize what is likely the simplest local solvers in each category to simplify the treatment and implementation. For a method without a setup phase, we will use Jacobi iterations and when more iterations are used, we will use incomplete LU factorization with zero fill-in (ILU-0) along with the multiscale operator stabilized with GMRES. Likewise, we have chosen to include correction functions since they previously have been used in a large body of literature on the MsFV method.

Although it is possible to use the approach described above to iterate to any desired accuracy, it may not be computationally feasible compared with other methods such as algebraic multigrid, which may achieve better performance if the target is to reduce the fine-scale residual to machine precision. In most workflows, however, one will typically seek computational savings by aborting the iteration of the multiscale pressure solution before it reaches machine precision accuracy or even just use the initial MsTPFA approximation, followed by a mass-conservative reconstruction. In the next section, we will therefore look at both the initial solution quality, the quality after a single cycle with a few Jacobi iterations, as well as convergence for Ms-ILU0-GMRES.

2.7. Connection to upscaling techniques

The MsTPFA method described herein can be seen as a multiscale realization of the standard local interface based transmissibility upscaling. It should be noted that while this paper is focused strictly on this specific coarse scale operator, it is possible to use the same framework to create multiscale-like methods from conventional upscaling techniques. As such, our work is similar to [32–35]. Further avenues of research would include incorporating generic, global solutions in the creation of the basis functions or by creating multipoint-like stencils by changing the local problems. Even though we have specifically chosen the coarse faces for the partition of unity and the local solutions to get a TPFA-like operator, another choice would be to include the corners to get a MPFA-like stencil.

3. Numerical results

The MsTPFA method has been implemented as an open-source prototype as part of the `msfvm` module in MRST [38,29]. To validate the method, we will consider several different test cases with both varying permeability and geometry. In all examples, we will compare the approximate multiscale solutions to the fine-scale reference solutions using scaled L_∞ and L_2 norms,

$$\|\boldsymbol{p}\|_\infty = \frac{\|\boldsymbol{p}_{fs} - \boldsymbol{p}_{ms}\|_\infty}{\|\boldsymbol{p}_{fs}\|_\infty}, \qquad \|\boldsymbol{p}\|_2 = \frac{\|\boldsymbol{p}_{fs} - \boldsymbol{p}_{ms}\|_2}{\|\boldsymbol{p}_{fs}\|_2}. \tag{32}$$

As the proposed multiscale formulation has the advantage of being very flexible with regards to grids, coarse grids that adapt to permeability features will be used when appropriate. To produce adapted grids, we will use METIS [39] configured with the transmissibilities of the fine-scale system as weights for the edge-cut minimization algorithm. We also allow the coarse blocks to vary by 75% in cell number to give the partitioning algorithm some freedom to capture fine-scale details. To give fairly regular blocks in regions of low contrast, the '-minconn' option was specified. In our experience, this is a simple and robust method for generating a coarse partition with a given number of primal blocks that provide a priori adaption to local features. This method is also applicable for cases where flow-adapted coarsening [40] is intractable because of grid size or when boundary conditions and/or well patterns may change during the simulation or between multiple forward simulations. Although it would be possible to also use information of the boundary conditions in the construction of the coarse grid, we will for simplicity not consider such coarse grids here.

3.1. SPE10

Model 2 from the 10th Comparative Solution Project [41] was originally designed as a challenging benchmark for upscaling method, and computing flow on subsets of this model has over the years been established as the de facto for multiscale methods. The model is part of a Brent sequence and contains two formations that are qualitatively different: the Tarbert formation represents a prograding near-shore environment in which the permeabilities follow a lognormal distribution, giving smoothly varying heterogeneities that most multiscale methods are capable of resolving quite well. The Upper Ness formation is fluvial with long high-permeable sand channels appearing in the west-east direction interbedded with low-permeable mudstone. The combination of long correlation lengths and strong abrupt changes makes the Upper Ness very challenging to resolve correctly.

We have run a large number of tests on 2D and 3D sub-samples of the SPE10 model. For brevity, we only report two examples that illustrate representative behavior of the MsTPFA method and highlight differences with the MsFV method.

Horizontal layers. As a first example, we will consider two horizontal 60 × 220 sub-samples: the top layer of the Tarbert formation (logical index $k = 1$ in the full model) and a highly channelized layer of the Ness formation ($k = 45$). In both cases, flow is driven by a pressure drop from the east to the west boundary. We consider two different coarse grids: a uniform 12 × 22 partition and a 264-block partition produced by METIS, giving an upscaling factor of 50. Fig. 7 shows the permeability fields, the coarse grids, and the MsTPFA approximate solutions before and after five iterations with a block-Jacobi smoother. The corresponding errors are reported in Table 1, while Fig. 8 reports the reduction in residual for a full multiscale iteration.

For the Tarbert layer, the coarse-scale operator is well behaved and adapting the coarse partition to structures in the fine-scale transmissibilities only reduces the initial error but does not affect the convergence of subsequent iterations. For the Upper Ness layer, we observe several regions with non-monotone pressure caused by weak multipoint connections in the coarse-scale stencil, which are induced because the basis functions will overlap at the coarse-grid vertices for tensor-product partitions, as mentioned in Section 2.6. Adapting the coarse grid to the transmissibilities in the fine grid gives partitions that do not follow the axial directions and hence have less overlap effects, which in turn diminishes the non-monotone behavior and significantly improves the accuracy of the coarse-scale stencil. As a result, the error and convergence that are comparable to the Tarbert layer. Similar behavior has been observed in many other experiments: rectangular partitions may introduce certain non-monotonicities for the MsTPFA method, and these are slightly stronger in 3D for non-unit aspect and anisotropy ratios, but can to a large extent be mitigated by using partitions that adapt to fine-scale transmissibilities.

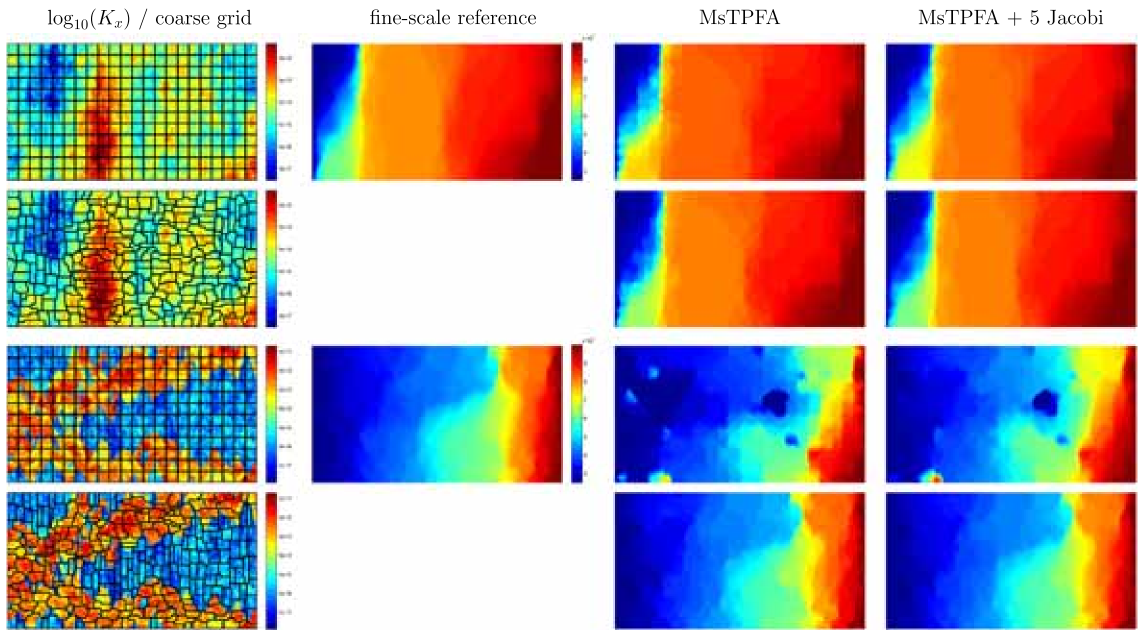


Fig. 7. Multiscale solutions for two horizontal sub-samples of the SPE10 data set.

Table 1
The error in the pressure solution for two horizontal subsets of SPE10 computed for two different coarse grids by a single multiscale cycle without smoothing and with five Jacobi iterations for both the MsTPFA and the MsFV methods.

Setup of model			Multiscale		Multiscale + Jacobi	
Permeability	Solver	Coarse grid	L_2	L_∞	L_2	L_∞
Tarbert	MsFV	uniform	0.0069	0.0586	0.0035	0.0239
Tarbert	MsTPFA	uniform	0.0735	0.2641	0.0450	0.1389
Tarbert	MsTPFA	adapted	0.0262	0.1741	0.0158	0.0798
Upper Ness	MsFV	uniform	0.9525	19.353	1.3696	11.227
Upper Ness	MsTPFA	uniform	0.3331	3.1365	0.2395	2.2114
Upper Ness	MsTPFA	adapted	0.0427	0.1415	0.0243	0.0577

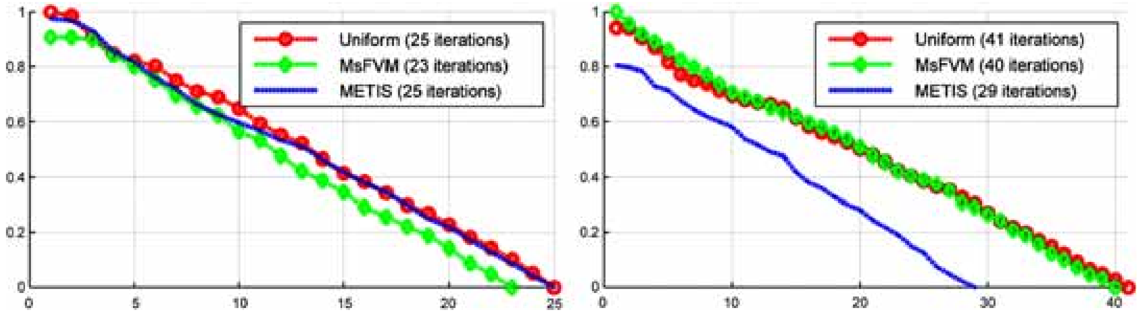


Fig. 8. Convergence of multiscale cycles to a tolerance of 10^{-8} on two horizontal sub-samples of the SPE10 data set. The y-axis is logarithmic and normalized to the largest initial residual error and the convergence of the final solution.

In Table 1 we have also included results computed by the MsFV method on the 12 × 22 partition for comparison. In our experience, the MsFV method is unusually accurate compared with other multiscale methods for problems with moderate heterogeneities, flow driven by a pressure drop in one of the axial directions, and Cartesian grids (with unit aspect ratios). The unusually low error that can be observed in such cases is explained by the fact that the reduced boundary conditions used to localize the computation of basis functions in the MsFV method are not very different from the true flow field.

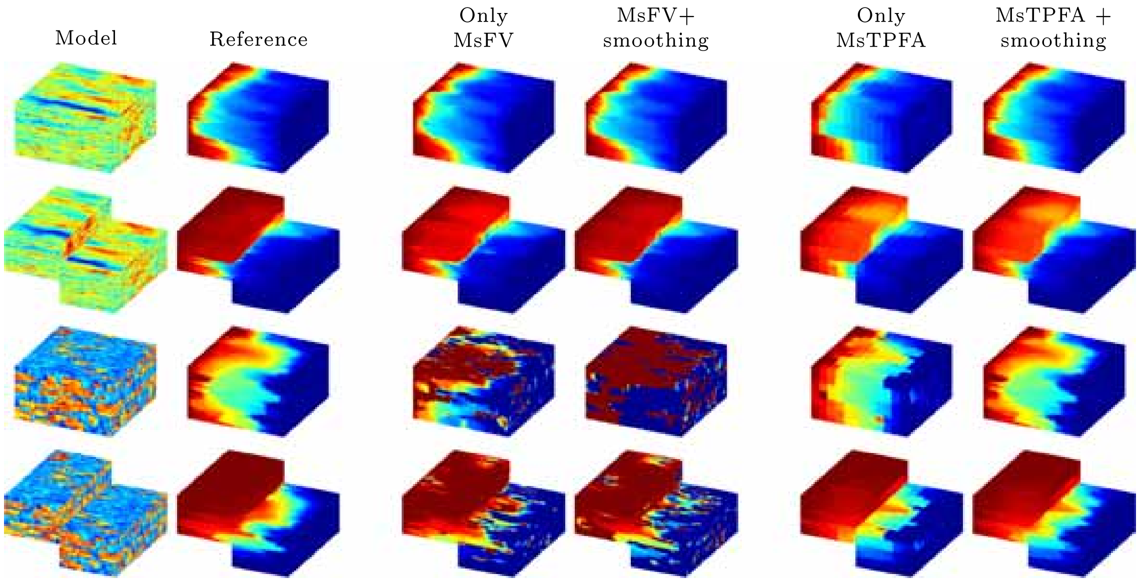


Fig. 9. Multiscale solutions computed by the MsFV and MsTPFA methods before and after smoothing for the 3D models using permeability sampled from the SPE10 data set.

The localization of the basis functions in the MsTPFA method is less accurate, and hence the error is larger than for the MsFV method. On more realistic flow scenarios, in which the flow is not essentially unidirectional, the reduced boundary conditions of the MsFV method will generally not be more representative than those of the MsTPFA method, and the accuracy of the two methods will be comparable. We have previously observed the same trend when comparing the MsFV and the MsMFE method, see e.g., [18]. For the Upper Ness sub-sample, and other strongly heterogeneously cases, MsFVs multipoint coarse-scale stencil will typically introduce strong unphysical oscillations that may prevent iterative versions of the method from converging properly, see [18,17,19,42–44,20,15].

3D box model, slip fault. In this example, which is taken from [15], we compare the MsTPFA method with the MsFV method on 3D box models with and without a single slip fault. Our current method for generating the wire-basket ordering [30,31] needed by MsFV is not yet sufficiently robust to always produce a sensible dual partition for arbitrary transmissibility-adapted primal grids generated by METIS, and hence we only consider uniform primal partitions. Fig. 9 reports a comparison of the MsFV and MsTPFA solutions before and after smoothing for four different 3D models. For the relatively smooth Tarbert formation, the MsFV method computes solutions that are qualitatively more accurate than those of the MsTPFA method, except for the small unphysical solutions that are difficult to spot in the plots. For the strongly heterogeneous Upper Ness formation, on the other hand, the MsFV method introduces large unphysical oscillations that are amplified by smoothing iterations, whereas the MsTPFA method is able to capture a qualitatively correct picture of the fine-scale solution. Altogether, this example is a good illustration of the purpose of the MsTPFA method: by sacrificing a certain accuracy for problems with smooth heterogeneity, one gains significantly in robustness for strongly heterogeneous problems.

3.2. Realistic bed model

One of the main sources of unstructured connections in reservoir modeling is pinch-outs, for which erosion or other geological features lead to inactive or highly degenerate cells. To get a test case with a large number of pinch-outs, we consider a highly detailed model designed to reproduce realistic, small-scale bedding structures on a scale much smaller than a single full reservoir simulation block. The model contains approximately 90 000 fine cells and has previously been used as test case for the MsMFE method [22] and the MsFV method [15]. Even though the domain is rectangular in appearance, almost every cell contains non-neighboring connections and degenerate features. The variance of the permeability distribution is for the most part modest, but intersecting layers of shale make the model especially challenging for the localization assumption in the MsFV method. This type of model is typically built as part of an upscaling workflow to determine effective directional permeability for a given lithofacies and to identify net pay below petrophysical log resolution. Hence, we use Dirichlet boundary conditions to impose a constant pressure drop in the horizontal direction.

We consider three different partition methods that all generate a coarse model with 54 blocks: (i) a simple uniform partition in index space as used for the MsFV method in [15]; (ii) a METIS partition as discussed above; and (iii) a METIS partition in which we do not allow coarse blocks that mix the low-permeable shale and the high-permeable background.

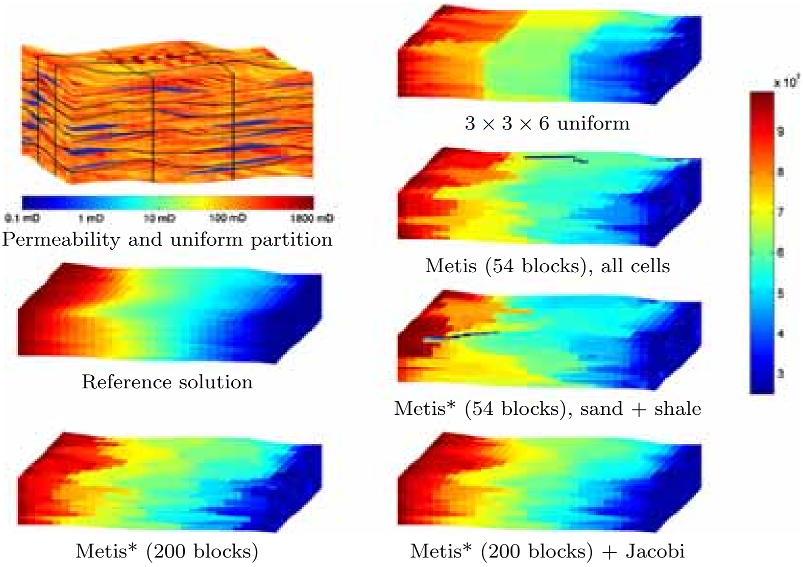


Fig. 10. Multiscale solutions computed on different partitions for a high-resolution, core-scale bedding model.

Table 2
Error in the pressure solution for a SBED model containing large amount of pinched and inactive cells for some coarse blocks. Here, METIS* refers to an alternate partition in which shales and sand are enforced as separate partitions.

Coarse grid	Blocks	MsTPFA		MsTPFA + Jacobi	
		L_2	L_∞	L_2	L_∞
Uniform	54	0.1048	0.9507	0.0722	0.4019
METIS	54	0.0967	0.9461	0.0744	0.5111
METIS*	54	0.1166	0.9859	0.0909	0.3849
METIS*	200	0.0899	0.9867	0.0665	0.2108
METIS*	1000	0.0448	1.0113	0.0273	0.2275
METIS*	5000	0.0202	0.6534	0.0109	0.0594

For partitions with many blocks, METIS has a tendency to create strongly convex blocks (e.g., consisting of a set of thin shale cells plus a column of sand cells), which will typically produce negative transmissibilities for the coarse system. The third approach is introduced to avoid creating such blocks and uses a straightforward modification of the transmissibility graph so that sand and shale cells are partitioned separately into smaller regions based on transmissibilities. Fig. 10 compares the initial multiscale solutions obtained for the three different partition strategies. Errors in L_2 and L_∞ norm before and after smoothing iterations are reported in Table 2. For comparison, the figure and table also include results obtained on finer resolutions for the third partition method. With uniform partition, the multiscale solution has a strong patchy behavior that cannot be rectified by smoothing iterations. Using METIS to partition the cells gives a qualitatively correct solution, except for a small region of unphysical pressure values which are efficiently removed by smoothing cycles. When partitioning sand and shale separately, almost one third of the 54 blocks are required to represent disconnected shale units, which leaves METIS less freedom to create a good partition of the sand volume in which most of the flow occurs. As a result, the corresponding multiscale solution has more grid artifacts than for the second strategy, but after smoothing the L_∞ error is lower than for the other methods.

The third strategy is more effective for a higher number of coarse blocks. To demonstrate this, we consider a wide range of coarsening degrees ranging from 54 to 5000 coarse blocks. The histogram in Fig. 11 confirms that the initial L_2 error decays with the size of the coarse blocks. The right-hand plot in Fig. 11 reports reduction in residual by an iterative Ms-ILU-GMRES solver. With only 54 blocks, the reduction in residual decays after the first few iterations, indicating that the multiscale coarse operator only contributes significantly to the initial reduction of the residual. With 5000 blocks on the other hand, the multiscale coarse operator contributes significantly to reduce the residual all the way to machine precision.

3.3. The Norne Field

For this example, we will use the corner-point grid from the simulation model of the Norne reservoir from the Norwegian Sea. The simulation model has a layered permeability distribution and contains faults, eroded cells and pinch-out leading

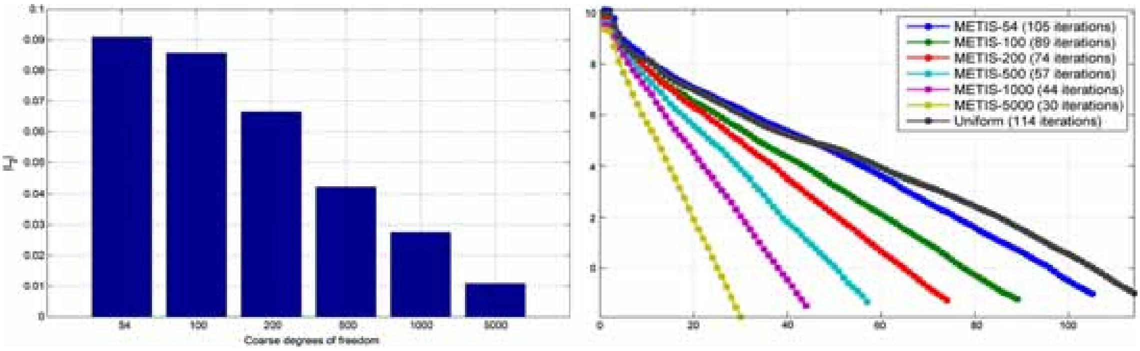


Fig. 11. The L_2 error for the bed model as a function of varying number of coarse blocks. The histogram shows the error of the initial multiscale solution, and the line plots show the reduction in the logarithm of the preconditioned L_2 residual for the corresponding Ms-ILU-GMRES solver.

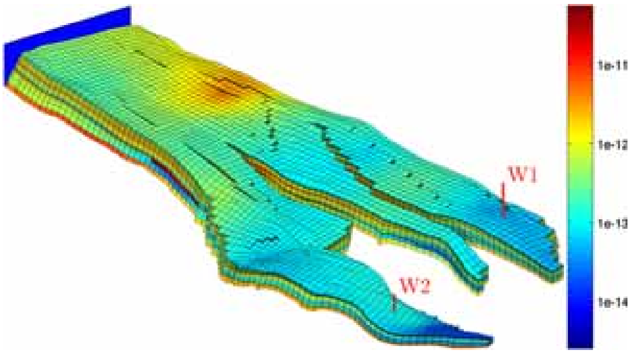


Fig. 12. The grid and permeability for the Norne Field case along with producer wells and boundary conditions.

Table 3
Error in the pressure solution for a realistic simulation model with geometry and permeability data from the Norne Field. Approximate solutions are compute for three different coarse grids by a single multiscale cycle without smoothing and with ten Jacobi iterations.

Coarse grid	MsTPFA		MsTPFA + Jacobi	
	L_2	L_∞	L_2	L_∞
Uniform	0.0782	0.7771	0.0554	0.3338
Fault-adapted	0.0599	0.9039	0.0387	0.1259
METIS	0.0662	0.9132	0.0402	0.0971

to non-neighboring connections as well as jumps in permeability. The grid contains approximately 45 000 fine cells, out of which 25% have either more or less than six faces, with the number of faces ranging from four to nineteen. The input data uses a description based on a structured topology, but once processed, the resulting grid has a non-structured topology and irregular geometry. To give flow across the entire domain, two vertical 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, see Fig. 12.

We will simultaneously consider three different coarse partitions that give approximately 200 coarse degrees of freedom: (i) a simple uniform partition in index space that takes neither geometry nor permeability into account, but gives an unstructured coarse grid because of inactive cells in the fine grid; (ii) a uniform partition but using geometry information to split blocks across faults; and (iii) a purely topological partition produced by METIS. Table 3 reports errors for the three partitions. The last two partitions introduce large errors near the boundaries of the coarse blocks and thus have large initial L^∞ errors. However, these errors can be efficiently removed by a few inexpensive Jacobi iterations. The uniform partition has lower initial L_∞ error, but since the coarse-scale operator is less accurate, the L_∞ error is significantly higher compared with the other two partitions after a single multiscale cycle. Similar results are reported for the MsFV method with the fault-adapted partition in [15]; for the uniform partition, the MsFV method failed to produce a solution, and for the METIS partition, generating a dual partition was not feasible.

To investigate the robustness of the multiscale operator, we use METIS to generate ten different coarse partitions, with number of blocks ranging from 10 to 5000, giving upscaling factors from 9 to 4500. The left plot in Fig. 13 reports the L_2

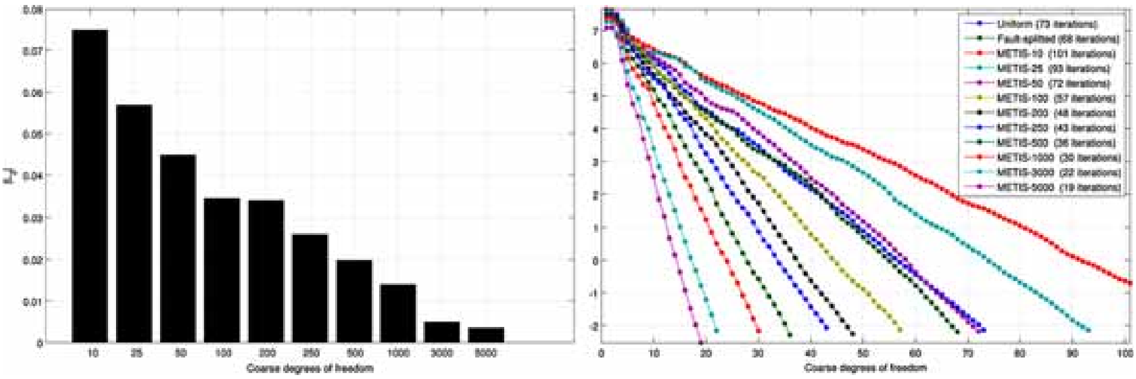


Fig. 13. The L_2 error for the Norne Field for varying number of coarse blocks. The histogram shows the error of the initial multiscale solution, and the line plots show the reduction in the logarithm of the preconditioned L_2 residual for the corresponding Ms-ILU-GMRES solver.

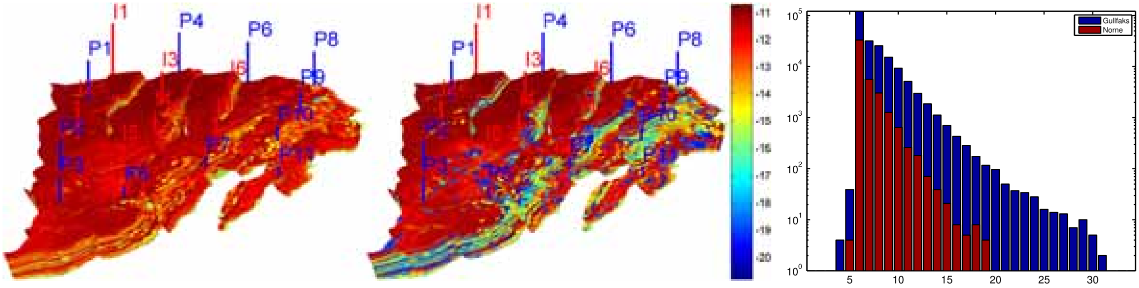


Fig. 14. An example of a structurally complex reservoir model: the Gullfaks Field. The two left-most plots show the reservoir geometry with horizontal and vertical permeability (log-scale) and wells colorized by type. The right plot shows a histogram of the number of faces per cell compared to the Norne model, which is significantly less structurally complex.

error of the initial multiscale solutions, whereas the right plot shows convergence the full Ms-ILU-GMRES iterative method. The convergence of the Ms-ILU-GMRES solver is similar to the results for the SBED example, demonstrating the same robustness and convergence even in the presence of faults and realistic geometry. For coarse partitions, the convergence of the method is very slow and from the kink in the residual plot one can infer that the multiscale coarse operator contributes little to the convergence after a few cycles when using less than one hundred blocks. Continued reduction of the residual is therefore left to the relatively inefficient ILU and GMRES iterations. However, each cycle has a very low computational cost and the first few cycles reduce the error significantly. Hence, one can use MsTPFA on a very coarse grid as an alternative to proxy models in workflows that do not require high simulation accuracy. On the opposite end, using more than one thousand blocks gives a very low initial error and rapid convergence, but the cost of each multiscale cycle is now significantly higher. Optimal efficiency is expected to lie somewhere in between, corresponding to an upscaling factor of a few hundred. Our prototype is not yet parallelized and fully optimized and this reason we refrain from a thorough study of efficiency.

The figure also reports the convergence history for the uniform partition and the uniform partition with splitting of blocks containing faults. For both, the reduction in the residual is comparable to that of the METIS partitions with approximately the same number of blocks for the first 5–10 iterations, but then decays significantly and follows the trend of the 50-block METIS partition. For the 200-block and 250-block METIS partitions, on the other hand, we observe no decay in the reduction rate, which demonstrates that using a permeability-adapted coarse partition gives a significantly better multiscale operator.

3.4. The Gullfaks Field

In the last example, we will demonstrate that the MsTPFA method is capable of handling models with a high structural complexity. To this end, we consider a simulation model of the Gullfaks Field, an oil and gas field in the Norwegian sector of the North Sea. The simulation model is represented using an $80 \times 100 \times 52$ corner-point grid in which 216 334 cells are active. The geometry of the Gullfaks model is highly irregular because of the large number of faults. Almost 44% of the cells have non-neighboring connections, and after the non-matching grid has been processed into a matching unstructured grid, the number of cell faces range from four to thirty-one as shown in Fig. 14. The permeability varies several orders of

Table 4
Errors in the pressure solution computed by the MsTPFA method for the Gullfaks model.

Coarse grid	No. blocks	MsTPFA		MsTPFA + Jacobi	
		L_2	L_∞	L_2	L_∞
Uniform	605	0.0581	0.3058	0.0518	0.2893
METIS	10	0.1049	0.3379	0.1001	0.2390
METIS	100	0.0519	0.3041	0.0429	0.2935
METIS	500	0.0324	0.2969	0.0262	0.2936
METIS	1000	0.0303	0.3016	0.0230	0.1769
METIS	10 000	0.0206	0.1994	0.0107	0.1070

magnitude and contains impermeable regions in the horizontal permeability, making this the absolutely most challenging test case in this paper.

To drive flow in the model, we consider a simple pattern of vertical wells distributed somewhat unrealistically throughout the model. The wells are controlled by bottom-hole pressure and the average injector pressure is set to 500 bars while the producers operate at an average pressure of 250 bars. The injectors and producers have per well variations in an order of 100 bars to make the pressure distribution more challenging to capture accurately.

As in the earlier models, we consider both a uniform partition as well as a variety of automatically generated partitions of increasing resolution to investigate systematic error reduction. The uniform coarse grid has local dimensions of $7 \times 7 \times 10$, giving an upscaling factor of approximately 350 once eroded and pinched cells have been removed. No special restrictions are added to the unstructured coarse partitions and we do not use correction functions. The results can be seen in Table 4 and Fig. 15. As can be seen from the figures and the table, the pressure field is easily resolved by the MsTPFA method, which gives high-quality pressure solutions for arbitrary degrees of coarsening. Using increased resolution of the coarse partition results in higher solution quality and fewer iterations required for the Ms-GMRES-ILU solver.

3.5. Time-dependent problems

The MsTPFA method and other methods like it are not primarily intended as general-purpose linear solvers for elliptic Laplace-type equations. Instead, the primary appeal lies in their ability to use non-converged pressure solutions to reconstruct conservative flux fields that can be coupled with a transport solver to solve time-dependent multiphase problems. To demonstrate this ability, we revisit the two SPE10 sublayers discussed above, adding a flux boundary condition that injects a total of one pore volume over a year. The model is initially filled with oil (5 cP viscosity) and is produced by injecting water (1 cP viscosity) along the east boundary. Quadratic relative permeabilities are assumed for both fluids.

When solving problems where coefficients change during the simulation, adaptive recomputation of basis functions [45] can be utilized to accelerate the simulation. The idea is to update basis functions only when the change in saturation, or in total mobility, inside a coarse block exceeds a certain prescribed tolerance. We will use a simple criterion: if the saturation in any cell associated with a given local solution has changed by more than 0.25 since the previous update, the whole local solution is updated.

Snapshots of the simulations are reported in Fig. 16. We are using the same number of coarse blocks as previously, without any iterations. As can be seen from the figure, the multiscale solutions are in a good agreement with the fine-scale reference solutions, demonstrating that the MsTPFA fluxes can be used to solve transport problems. By plotting the number of times a cell is updated during the simulation, as shown in Fig. 17, we see that basis functions are updated more often in regions of high flow. For the Ness case, in particular, basis functions are hardly updated in low-permeability regions since these are not swept by the water front.

4. Concluding remarks

In this paper, we have presented a novel multiscale two-point flux-approximation (MsTPFA) method that relies on algebraic manipulations of discrete linear systems arising from standard fine-scale, finite-volume discretizations. The method is based on much of the same ideas as the multiscale finite-volume (MsFV) method and can utilize key technologies developed for this method, including reconstruction of conservative fine-scale fluxes and use of correction functions to more accurately capture effects from gravity, wells, etc. that are not accounted for in the basis functions. Moreover, the method can be cast within iterative frameworks that have been developed for the MsFV method to systematically drive fine-scale residuals towards machine precision. The main advantage of such an iterative framework compared with multigrid methods is that the iteration can be aborted at any step to reconstruct conservative fine-scale fluxes.

Two important aspects, the construction of coarse partitions and the structure of the coarse-scale system, distinguish the MsTPFA from the MsFV method and makes the former more flexible and robust. To localize basis functions, the MsFV method uses reduced, compatible boundary conditions that are vulnerable to strong heterogeneities. As a result, the method forms multipoint coarse-scale systems that may produce strongly non-monotone solutions. The MsTPFA method, on the other hand, uses a partition of unity derived by solving simple flow problems (tracer partitions) to form a coarse-scale system that is virtually a two-point discretization that computes almost monotone pressure solutions. This means that

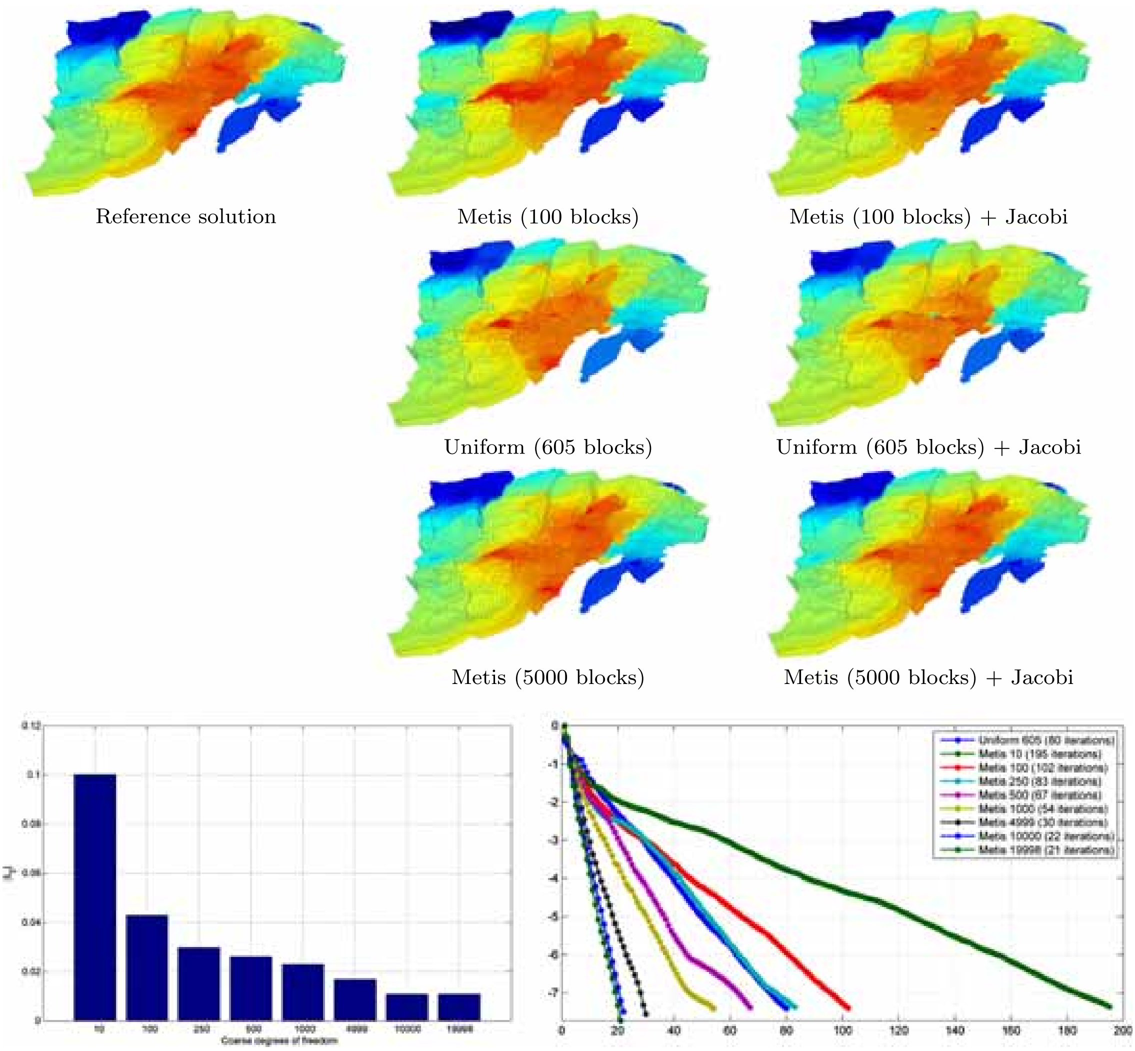


Fig. 15. Multiscale solutions computed on different partitions for the Gullfaks model and the corresponding convergence plots. The histogram shows the error of the initial multiscale solution, and the line plots show the reduction in the logarithm of the preconditioned L_2 residual (scaled by the largest initial residual) for the corresponding Ms-ILU-GMRES solver.

while the MsTPFA method is not as accurate as the MsFV method on smooth heterogeneities, it is significantly more robust on highly heterogeneous media with non-unit anisotropy and aspect ratios, which is particularly advantageous when the method is used as part of an iterative framework in combination with an inexpensive smoother. If a good starting point is provided for the smoother, high-quality solutions can be obtained within a few multiscale cycles even for problems with strong heterogeneities. As a simple rule of thumb, we suggest to use the MsTPFA method unless you know that the MsFV method will produce a stable coarse-scale operator, which, for instance, is the case for problems that are moderately heterogeneous and described on a Cartesian grid.

Secondly, while the MsFV method requires a compatible primal–dual partition to compute basis/correction functions and generate coarse-scale systems, the MsTPFA method is based on a single coarse partition and hence shares the flexibility that has previously been reported for the multiscale mixed finite-element method. The use of a single coarse partition makes the MsTPFA method simple to implement for general unstructured grids. Extensive numerical tests on stratigraphic grids with erosion, pinch-outs, faults, and other types of degeneracies and nonconformities show that the method is very robust with regard to the choice of coarse partitions and can produce approximate solutions that resemble the fine-scale reference for almost arbitrary degrees of coarsening. The method may still have grid effects that reduce the quality of the approximate solutions. However, such grid effects, as well as monotonicity problems, can be significantly reduced by using

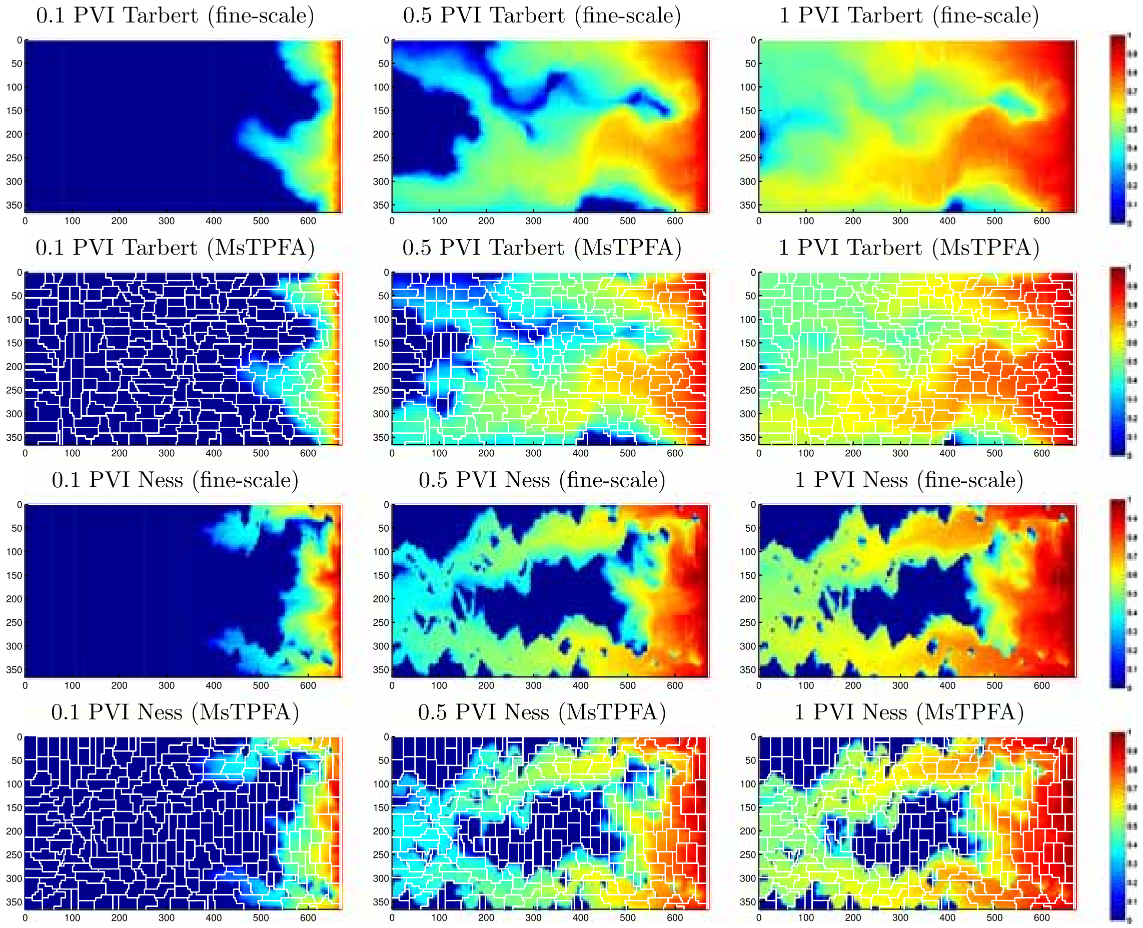


Fig. 16. Multiscale solutions for a waterflood on two horizontal sub-samples of the SPE10 data set (Layers 1 and 45, respectively).

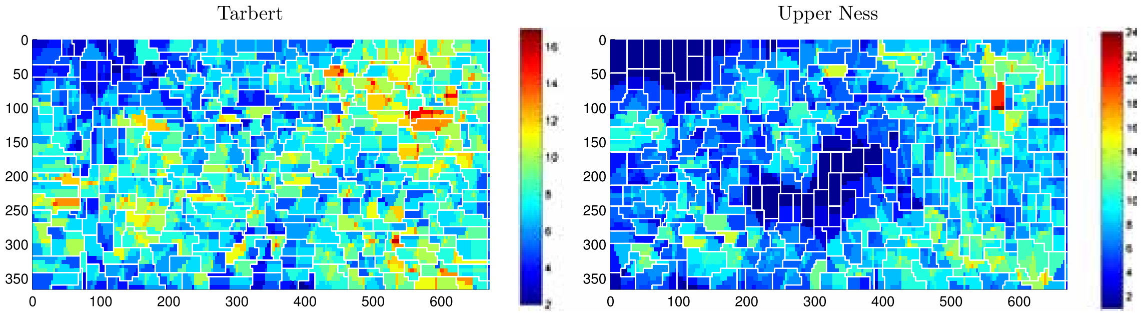


Fig. 17. The number of basis updates per fine cell for the MsTPFA-simulations reported in Fig. 16.

coarse partitions that adapt to the geometry and heterogeneities of the medium. Applying such grids to the multiscale finite-volume method, while simultaneously improving the coarse-scale operator, is a subject of ongoing research.

Finally, although the details are not yet worked out, we believe that the tracer partition idea can be extended to create other prolongation operators corresponding to multipoint coarse-scale discretizations. For Cartesian grids, the obvious starting point is to subdivide faces into multiple patches and create partition-of-unity functions that reflect the desired multipoint connections. For unstructured coarse and fine grids, the extension is less obvious.

Acknowledgements

The authors thank Jostein R. Natvig, Halvor Møll Nilsen, and Stein Krogstad for helpful discussions. Parts of the research has been funded by Schlumberger Information Solutions and the Research Council of Norway under grant No. 226035. The authors would also like to thank Statoil (operator of the Norne field) and its license partners ENI and Petoro for the release of the Norne data.

References

[1] Y. Efendiev, T.Y. Hou, Multiscale Finite Element Methods, Surveys and Tutorials in the Applied Mathematical Sciences, vol. 4, Springer Verlag, New York, 2009.

[2] 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, http://dx.doi.org/10.1016/S0021-9991(03)00075-5.

[3] Z. Chen, T. Hou, A mixed multiscale finite element method for elliptic problems with oscillating coefficients, Math. Comput. 72 (2003) 541–576, http://dx.doi.org/10.1090/S0025-5718-02-01441-2.

[4] J.E. Aarnes, On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, Multiscale Model. Simul. 2 (3) (2004) 421–439, http://dx.doi.org/10.1137/030600655.

[5] S.H. Lee, H. Zhou, H.A. Tchelepi, Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations, J. Comput. Phys. 228 (24) (2009) 9036–9058, http://dx.doi.org/10.1016/j.jcp.2009.09.009.

[6] I. Lunati, P. Jenny, Multiscale finite-volume method for compressible multiphase flow in porous media, J. Comput. Phys. 216 (2) (2006) 616–636, http://dx.doi.org/10.1016/j.jcp.2006.01.001.

[7] I. Lunati, P. Jenny, Multiscale finite-volume method for density-driven flow in porous media, Comput. Geosci. 12 (3) (2008) 337–350, http://dx.doi.org/10.1007/s10596-007-9071-9.

[8] H. Zhou, H.A. Tchelepi, Operator-based multiscale method for compressible flow, SPE J. 13 (2) (2008) 267–273, http://dx.doi.org/10.2118/106254-PA.

[9] H. Hajibeygi, G. Bonfigli, M.A. Hesse, P. Jenny, Iterative multiscale finite-volume method, J. Comput. Phys. 227 (19) (2008) 8604–8621, http://dx.doi.org/10.1016/j.jcp.2008.06.013.

[10] H. Hajibeygi, P. Jenny, Adaptive iterative multiscale finite volume method, J. Comput. Phys. 230 (3) (2011) 628–643, http://dx.doi.org/10.1016/j.jcp.2010.10.009.

[11] I. Lunati, S.H. Lee, An operator formulation of the multiscale finite-volume method with correction function, Multiscale Model. Simul. 8 (1) (2009) 96–109, http://dx.doi.org/10.1137/080742117.

[12] I. Lunati, M. Tyagi, S.H. Lee, An iterative multiscale finite volume algorithm converging to the exact solution, J. Comput. Phys. 230 (5) (2011) 1849–1864, http://dx.doi.org/10.1016/j.jcp.2010.11.036.

[13] S.H. Lee, C. Wolfsteiner, H. Tchelepi, Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three phase flow with gravity, Comput. Geosci. 12 (3) (2008) 351–366, http://dx.doi.org/10.1007/s10596-007-9069-3.

[14] H. Hajibeygi, H.A. Tchelepi, Compositional multiscale finite-volume formulation, SPE J. 19 (2) (2014) 316–326, http://dx.doi.org/10.2118/163664-PA.

[15] O. Møyner, K.-A. Lie, The multiscale finite-volume method on stratigraphic grids, SPE J. (2014), http://dx.doi.org/10.2118/163649-PA.

[16] O. Møyner, Multiscale finite-volume methods on unstructured grids, Master's thesis, Norwegian University of Science and Technology, Trondheim, 2012, http://daim.idi.ntnu.no/masteroppgave?id=7377.

[17] I. Lunati, P. Jenny, Treating highly anisotropic subsurface flow with the multiscale finite-volume method, Multiscale Model. Simul. 6 (1) (2007) 308–318, http://dx.doi.org/10.1137/050638928.

[18] V. Kippe, J.E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Comput. Geosci. 12 (3) (2008) 377–398, http://dx.doi.org/10.1007/s10596-007-9074-6.

[19] M.A. Hesse, B.T. Mallison, H.A. Tchelepi, Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations, Multiscale Model. Simul. 7 (2) (2008) 934–962, http://dx.doi.org/10.1137/070705015.

[20] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Algebraic multiscale solver for flow in heterogeneous porous media, J. Comput. Phys. 259 (2014) 284–303, http://dx.doi.org/10.1016/j.jcp.2013.11.024.

[21] J.E. Aarnes, S. Krogstad, K.-A. Lie, A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids, Multiscale Model. Simul. 5 (2) (2006) 337–363, http://dx.doi.org/10.1137/050634566.

[22] J.E. Aarnes, S. Krogstad, K.-A. Lie, Multiscale mixed/mimetic methods on corner-point grids, Comput. Geosci. 12 (3) (2008) 297–315, http://dx.doi.org/10.1007/s10596-007-9072-8.

[23] J.R. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, S.K. Khataniar, Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs, SPE J. 16 (4) (2011) 880–888, http://dx.doi.org/10.2018/119132-PA.

[24] F.O. Alpak, M. Pal, K.-A. Lie, A multiscale method for modeling flow in stratigraphically complex reservoirs, SPE J. 17 (4) (2012) 1056–1070, http://dx.doi.org/10.2118/140403-PA.

[25] M. Pal, S. Lamine, K.-A. Lie, S. Krogstad, Multiscale method for simulating two and three-phase flow in porous media, in: SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18–20 February 2013, 2013, SPE 163669-MS.

[26] S. Krogstad, K.-A. Lie, B. Skaflestad, Mixed multiscale methods for compressible flow, in: Proceedings of ECMOR XIII – 13th European Conference on the Mathematics of Oil Recovery, EAGE, Biarritz, France, 2012, http://www.earthdoc.eage.org/publication/publicationdetails/?publication=62218.

[27] S. Krogstad, A sparse basis POD for model reduction of multiphase compressible flow, in: 2011 SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 21–23 February 2011, 2011.

[28] S. Krogstad, K.-A. Lie, H.M. Nilsen, J.R. Natvig, B. Skaflestad, J.E. Aarnes, A multiscale mixed finite-element solver for three-phase black-oil flow, in: SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009, 2009.

[29] K.-A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H. Nilsen, B. Skaflestad, Open-source MATLAB implementation of consistent discretisations on complex grids, Comput. Geosci. 16 (2012) 297–322, http://dx.doi.org/10.1007/s10596-011-9244-4.

[30] B.F. Smith, P.E. Bjørstad, W.D. Gropp, Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, Cambridge, 1996.

[31] J. Wallis, H.A. Tchelepi, Apparatus, method and system for improved reservoir simulation using an algebraic cascading class linear solver, U.S. Patent No. 7684967, 2010.

[32] M. Gerritsen, J.V. Lambers, Integration of local–global upscaling and grid adaptivity for simulation of subsurface flow in heterogeneous formations, Comput. Geosci. 12 (2) (2008) 193–208, http://dx.doi.org/10.1007/s10596-007-9078-2.

[33] J.V. Lambers, M.G. Gerritsen, B.T. Mallison, Accurate local upscaling with variable compact multipoint transmissibility calculations, Comput. Geosci. 12 (3) (2008) 399–416, http://dx.doi.org/10.1007/s10596-007-9068-4.

[34] T. Chen, M.G. Gerritsen, L.J. Durlofsky, J.V. Lambers, Adaptive local–global VCMP methods for coarse-scale reservoir modeling, in: SPE Reservoir Simulation Symposium, The Woodlands, Texas, 2–4 February 2009, 2009.

[35] T. Chen, M.G. Gerritsen, J.V. Lambers, L.J. Durlofsky, Global variable compact multipoint methods for accurate upscaling with full-tensor effects, Comput. Geosci. 14 (1) (2010) 65–81, http://dx.doi.org/10.1007/s10596-009-9133-2.

[36] J.R. Natvig, K.-A. Lie, B. Eikemo, I. Berre, An efficient discontinuous Galerkin method for advective transport in porous media, Adv. Water Resour. 30 (12) (2007) 2424–2438, http://dx.doi.org/10.1016/j.advwatres.2007.05.015.

[37] H. Hajibeygi, Iterative multiscale finite volume method for multiphase flow in porous media with complex physics, Ph.D. thesis, ETH, Zurich, 2011, http://dx.doi.org/10.3929/ethz-a-006696714.

[38] The MATLAB reservoir simulation toolbox, version 2013a, http://www.sintef.no/MRST/, Apr. 2013.

[39] Metis – serial graph partitioning and fill-reducing matrix ordering, http://glaros.dtc.umn.edu/gkhome/views/metis, 2012.

[40] K.-A. Lie, J.R. Natvig, S. Krogstad, Y. Yang, X.-H. Wu, Grid adaptation for the Dirichlet–Neumann representation method and the multiscale mixed finite-element method, Comput. Geosci. (2014), http://dx.doi.org/10.1007/s10596-013-9397-4.

[41] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, SPE Reserv. Eval. Eng. 4 (2001) 308–317, http://dx.doi.org/10.2118/72469-PA, http://www.spe.org/csp/.

[42] J.M. Nordbotten, E. Keilegavlen, A. Sandvin, Mass conservative domain decomposition for porous media flow, in: R. Petrova (Ed.), Finite Volume Method–Powerful Means of Engineering Design, InTech Europe, Rijeka, Croatia, 2012, pp. 235–256.

[43] H. Zhou, H.A. Tchelepi, Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, SPE J. 17 (2) (2012) 523–539, http://dx.doi.org/10.2118/141473-PA.

[44] A. Sandvin, E. Keilegavlen, J.M. Nordbotten, Auxiliary variables for 3d multiscale simulations in heterogeneous porous media, J. Comput. Phys. 238 (2013) 141–153, http://dx.doi.org/10.1016/j.jcp.2012.12.016.

[45] P. Jenny, S.H. Lee, H.A. Tchelepi, Adaptive multiscale finite-volume method for multiphase flow and transport in porous media, Multiscale Model. Simul. 3 (1) (2004) 50–64, http://dx.doi.org/10.1137/030600795.

Paper IV

Construction of Multiscale Preconditioners on Stratigraphic Grids

Olav Møyner

In proceedings of the 14th European Conference on the Mathematics of Oil Recovery, 2014, Catania, Sicily, Italy.
DOI: 10.3997/2214-4609.20141775

Is not included due to copyright

Paper V

A Multiscale Restriction-Smoothed Basis Method for High Contrast Porous Media Represented on Unstructured Grids

Olav Møyner and Knut-Andreas Lie

Journal of Computational Physics, volume 304, pp. 46-71, 2016
DOI: 10.1016/j.jcp.2014.07.003

Journal of Computational Physics 304 (2016) 46–71

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids

Olav Møyner [*], Knut-Andreas Lie

SINTEF ICT, Department of Applied Mathematics, P.O. Box 124, Blindern, NO-0314 Oslo, Norway

ARTICLE INFO

Article history:
Received 9 May 2015
Received in revised form 2 October 2015
Accepted 5 October 2015
Available online 13 October 2015

Keywords:
Multiscale
Conservative
Reservoir
Multiphase
Porous media
Faults

ABSTRACT

A wide variety of multiscale methods have been proposed in the literature to reduce runtime and provide better scaling for the solution of Poisson-type equations modeling flow in porous media. We present a new multiscale restricted-smoothed basis (MsRSB) method that is designed to be applicable to both rectilinear grids and unstructured grids. Like many other multiscale methods, MsRSB relies on a coarse partition of the underlying fine grid and a set of local prolongation operators (multiscale basis functions) that map unknowns associated with the fine grid cells to unknowns associated with blocks in the coarse partition. These mappings are constructed by restricted smoothing: Starting from a constant, a localized iterative scheme is applied directly to the fine-scale discretization to compute prolongation operators that are consistent with the local properties of the differential operators.

The resulting method has three main advantages: First of all, both the coarse and the fine grid can have general polyhedral geometry and unstructured topology. This means that partitions and good prolongation operators can easily be constructed for complex models involving high media contrasts and unstructured cell connections introduced by faults, pinch-outs, erosion, local grid refinement, etc. In particular, the coarse partition can be adapted to geological or flow-field properties represented on cells or faces to improve accuracy. Secondly, the method is accurate and robust when compared to existing multiscale methods and does not need expensive recomputation of local basis functions to account for transient behavior: Dynamic mobility changes are incorporated by continuing to iterate a few extra steps on existing basis functions. This way, the cost of updating the prolongation operators becomes proportional to the amount of change in fluid mobility and one reduces the need for expensive, tolerance-based updates. Finally, since the MsRSB method is formulated on top of a cell-centered, conservative, finite-volume method, it is applicable to any flow model in which one can isolate a pressure equation. Herein, we only discuss single and two-phase incompressible models. Compressible flow, e.g., as modeled by the black-oil equations, is discussed in a separate paper.

1. Introduction

The general movement of fluids in a hydrocarbon reservoir is induced by global forces like gravity and pressure differentials. The micro-scale displacement, however, is determined by small-scale flow paths throughout highly heterogeneous

[*] Corresponding author.
E-mail addresses: olav.moyner@sintef.no (O. Møyner), knut-andreas.lie@sintef.no (K.-A. Lie).

http://dx.doi.org/10.1016/j.jcp.2015.10.010
0021-9991/© 2015 Elsevier Inc. All rights reserved.

porous rocks. Flow modeling therefore needs to take into account processes taking place on a wide range of spatial and temporal scales. Resolving all these scales using a single high-resolution grid is not computationally tractable. Instead, the traditional approach is to use upscaling or homogenization techniques to develop effective parameters that represent subscale behavior in an averaged sense on a coarser scale. Such methods have proved to be very effective for problems with scale separation. However, porous rocks seldom exhibit clear scale separation and upscaling techniques are therefore not as robust and accurate as one would wish. Effective properties are generally process dependent, and because one needs to assume a specific set of localization conditions to compute effective properties, upscaling techniques tend to only produce reliable results for a limited range of flow scenarios.

In an attempt to overcome some of the limitations of upscaling methods, so-called multiscale discretization methods have been proposed over the past two decades to solve second-order elliptic equations with strongly heterogeneous coefficients [1]. This includes methods such as the generalized finite-element methods [2], finite-element methods [3], numerical-subgrid upscaling [4,5], multiscale mixed finite-element methods [6,7], multiscale finite-volume methods [8], mortar mixed finite-element methods [9], and multiscale mimetic methods [10], to name a few. The key idea of all these methods is to construct a set of prolongation operators (or basis functions) that map between unknowns associated with cells of the fine geo-cellular grid and unknowns on a coarser grid used for dynamic simulation. The prolongation operators are computed numerically by solving localized flow problems, much in the same way as for flow-based upscaling methods, but unlike effective parameters, the multiscale basis functions have subscale resolution. The result is that local fine-scale variations can be systematically and correctly accounted for when constructing a reduced coarse-scale problem to study the macro-scale displacement driven by global forces. There is a large body of literature that develops such multiscale methods and studies their mathematical and numerical properties for idealized and simplified problems like the variable-coefficient Poisson equation formulated on Cartesian box geometry. However, to provide value for commercial applications, these methods also need to be developed so that they can handle the complexity in flow physics and geological description seen in real-life simulation models. Over the past decade, there have primarily been two main developments in this direction, focusing on the multiscale finite-volume (MsFV) method [8] and the multiscale mixed finite-element (MsMFE) method [6,11,12].

Research on the MsFV method has mainly focused on extending the method from incompressible flow to realistic flow physics [13–18] and on developing iterative approaches that ensure that the method converges to the correct solution of the underlying fine-scale discretization [16,19–21]. However, with a few notable exceptions [22–26], the MsFV method has so far only been studied on grids with a Cartesian topology. (An alternative finite-volume formulation is also discussed in [27].) Such grids are highly desirable in terms of accuracy, efficiency, and robustness of the numerical discretizations and solvers, and modeling approaches used in industry are therefore predominantly structured in a global sense. However, to accurately account for structural features like faults, joints, and deformation bands and stratigraphic characteristics like channels, lobes, clinoforms, and shale/mud drapes, unstructured connections are introduced locally and cell geometries tend to be (highly) skewed or degenerate. Similarly, unstructured connections may be introduced by local grid refinement, e.g., in the near-well zones. The challenge in extending the MsFV method to realistic stratigraphic grids, or in the more general sense to grids with fully unstructured topologies, lies in the underlying primal–dual coarse partition. The MsFV method computes basis functions on a dual partition to define transmissibilities in a multi-point coarse-scale discretization. Approximate solutions computed from these basis functions is then used to define boundary conditions for another set of flow problems on the primal partition to reconstruct conservative fine-scale fluxes. We have previously demonstrated that compatible primal–dual partitions can be generated for grids with degenerate cells and unstructured topologies if these grids are not too irregular [25]. However, the coarsening process is difficult to automate in a robust manner, and so far our most advanced algorithm is only able to provide semi-structured partitions for a limited range of coarsening factors. It is also well known that highly contrasted media and large anisotropy ratios may introduce strong non-monotonicities that are hard to get rid of in the iterative stages of the method [21].

For the MsMFE method, on the other hand, the main focus has been on making the method as geometrically flexible as possible and developing coarsening strategies that semi-automatically adapt to barriers, channels, faults, and wells in a way that ensures good accuracy for a chosen level of coarsening. The resulting method can efficiently predict flow patterns that are qualitatively correct for highly heterogeneous and geologically complex reservoir models under the assumption of incompressible flow [28–31]. The method has also been extended towards realistic compressible flow physics [32,33], but this has proved difficult to achieve in a fully robust manner because of the inherent assumption of a pressure equation written on mixed form. In a recent work [34], we presented a fully algebraic finite-volume framework that combines the best features of the MsFV and MsMFE methods and developed one specific method that mimics a coarse-scale two-point stencil by using numerically generated partition-of-unity functions to glue together elementary flow solutions associated with interfaces between coarse blocks. The resulting MsTPFA method generally produces high-quality approximate solutions for complex industry-standard grids with high aspect ratios and unstructured connections and can easily be extended to incorporate realistic flow physics. However, the method admittedly requires some intricate details to define and compute the partition-of-unity functions. Herein, we present a new and quite different multiscale formulation that offers the same robustness and flexibility as MsTPFA, but is much simpler to implement and gives very accurate interpolation.

The new method constructs mappings based on restricted smoothing: Starting from initial prolongation operators that are defined as the characteristic functions of each coarse block (i.e., equal unity inside the block and zero outside), a localized iterative scheme is applied directly to the fine-scale discretization to modify the prolongation operators so that they become increasingly consistent with the local properties of the differential operators. The use of weighted Jacobi smoothing on interpolation operators have been used with a large degree of success in the algebraic multigrid (AMG) community where fast coarsening is combined with simple operators constructed via one or two smoothing steps [35–38] as an inexpensive alternative to the interpolation operators used in standard AMG [39]. Many high performance multigrid solvers support smoothed aggregation as a strategy for large, complex problems [40,41] due to the inexpensive coarsening and interpolation strategies. A series of numerical experiments show that the new MsRSB method gives highly accurate prolongation operators for a wide variety of block shapes, e.g., including blocks that adapt to complex geological features in real-world models. Moreover, whereas methods like MsFV, MsMFE, and MsTPFA recompute the prolongation operator locally when faced with mobility changes in the underlying grid, the new method just continues the iteration until the operators are sufficiently smooth. This way, the cost of updating the prolongation operator becomes proportional to the amount of change in fluid mobility, eschewing the typical tolerance based updates. The formulation is algebraic and can be applied directly to linear systems, possibly in combination with existing multiscale techniques such as local stages and iterative cycles [21]. Through a series of numerical experiments, which include the well-known SPE 10 data and grid and petrophysical properties from two Norwegian oil fields, we validate our new MsRSB method and show that it is robust and efficient for single-phase and multiphase incompressible flow models. In a companion paper [42] we discuss how to extend the method to compressible flow problems and demonstrate that it provides one order-of-magnitude speedup compared to a fully-implicit simulator with the constrained pressure residual (CPR) preconditioner and algebraic multiscale preconditioner for compressible water-injection cases. [43] reports a comparison of the MsFV, MsTPFA, and MsRSB methods used as iterative solvers and shows that the MsRSB either performs equally well or clearly outperforms the other two methods for the studied test cases.

2. Model problems

Multiscale methods, as discussed herein, are designed to efficiently compute the approximate action of second-order elliptic differential operators of the form $\nabla \cdot \mathbf{K}(\boldsymbol{x})\nabla$, where the coefficient $\mathbf{K}(\boldsymbol{x})$ may exhibit orders of magnitude variations over short distances and contain short, intermediate, and long-range correlations. This operator primarily determines the pressure distribution, but may also govern temperature in thermal models.

2.1. Single-phase flow

To introduce the multiscale method and investigate its spatial approximation properties, it is sufficient to consider a incompressible single-phase flow in the absence of gravity, which is modeled by the variable-coefficient Poisson equation,

$$-\nabla \cdot (\mathbf{K}(\boldsymbol{x})\nabla p(\boldsymbol{x})) = q(\boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^d, \qquad \mathbf{K}(\boldsymbol{x}) \in \mathbb{R}^d \times \mathbb{R}^d, \tag{1}$$

where p is the fluid pressure, $\mathbf{K}$ is the permeability, and q denotes source terms. We discretize this equation using a standard finite-volume scheme,

$$\sum_j v_{ij} = q_i, \qquad v_{ij} = -T_{ij}(p_i - p_j), \tag{2}$$

where the transmissibility T_{ij} is associated with the interface between each pair of two cells i and j and defines a two-point flux approximation to the flux across this interface. For a Cartesian grid in 3D, (2) gives the standard seven-point finite-difference stencil. The resulting linear system

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{q} \tag{3}$$

is weakly diagonally dominant because each equation represents volume conservation over a single cell.

2.2. Multiphase flow

The basic model for multiphase flow consists of conservation of mass and Darcy's law for each phase α,

$$\partial_t\big(\phi\rho_\alpha S_\alpha\big) + \nabla\big(\rho_\alpha \vec{v}_\alpha\big) = \rho_\alpha q_\alpha, \qquad \vec{v}_\alpha = -\lambda_\alpha \mathbf{K}\big(\nabla p_\alpha - \rho_\alpha g \nabla z\big) \tag{4}$$

Here, ϕ denotes porosity, g is the gravity constant, and z the coordinate in the vertical direction, whereas S_α is the saturation (volume fraction) and $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$ the mobility of phase α, where $k_{r\alpha}$ is the relative permeability and μ_α the viscosity of the phase. This model has more unknowns than equations and we must therefore specify an additional closure relationship for the saturations, $\sum_\alpha S_\alpha = 1$, as well as relationships for the phase pressures that express the individual capillary pressures as know functions of fluid saturations.

In the following, we only consider incompressible two-phase flow. There are several ways one can choose primary variables and reorganize the resulting system of equations to express it as an elliptic equation for flow (pressure and fluxes) and a hyperbolic equation for fluid transport. Herein, we use the pressure and saturation of the wetting phase as our primary unknowns, giving the pressure equation

$$-\nabla\cdot\big(\lambda\mathbf{K}\nabla p_w\big)=q_n+q_w-\nabla\big[\lambda_n\mathbf{K}\nabla p_{cnw}+\mathbf{K}(\lambda_n\rho_n+\lambda_w\rho_w)g\nabla z\big], \tag{5}$$

and the transport equation

$$\phi\partial_t S_w+\nabla\cdot\Big(f_w\big[\vec{v}+\lambda_n\mathbf{K}(\rho_w-\rho_n)g\nabla z\big]\Big)-\nabla\cdot\big(\lambda_n\mathbf{K}\nabla p_{cnw}\big)=q_w, \tag{6}$$

where we have introduced the total Darcy velocity $\vec{v}=\vec{v}_w+\vec{v}_n$, the capillary pressure $p_{cnw}=p_n-p_w$, the total mobility $\lambda=\lambda_w+\lambda_n$, and the fractional flow function $f_w=\lambda_w/(\lambda_w+\lambda_n)$. The three latter are known functions of S_w.

To solve the system (5)–(6), we use a sequential procedure, in which we first solve (5) to compute p_w and $\vec{v}$ and then hold these constant while advancing (6) one time step. For spatial discretization of (5), we use the same two-point finite-volume scheme as for the single-phase equation, extended with upstream weighting of all terms that depend on saturation. (Note, however, that we could equally well have used a multipoint flux-approximation for the spatial discretization.) For (6), we use potential ordering to determine the upstream weighting [44] for the mobilities on the faces $\lambda_\alpha=k_{r\alpha}/\mu_\alpha$, which in most situations coincides with the upstream weighting used in the pressure solver. To allow for longer time steps, we use an implicit temporal discretization, giving a nonlinear discrete system that is solved with Newton's method. If necessary, one can also iterate the pressure and transport steps to ensure that the residual of the combined discrete system is below a prescribed threshold. However, for the cases considered later in the paper, we have not found this necessary.

3. Multiscale formulation

As explained above, our definition of a multiscale method starts from a fine grid $\{\Omega_i\}_{i=1}^n$ and a coarse partition that defines a coarse grid $\{\bar{\Omega}_j\}_{j=1}^m$ so that each fine cell in Ω belongs to only one coarse block in $\bar{\Omega}$. We then define a numerical prolongation operator, $P:\{\bar{\Omega}_j\}\to\{\Omega_i\}$, that maps quantities associated with the coarse blocks to quantities associated with the fine cells. Likewise, we define a restriction operator as the analogous map going the other way $R:\{\Omega_i\}\to\{\bar{\Omega}_j\}$. In the implementation, these operators are represented as sparse matrices of size $n\times m$ and $m\times n$, respectively. If we now let $\boldsymbol{p}_c$ denote a pressure computed on the coarse grid, we can find a fine-scale approximate pressure $\boldsymbol{p}_f$ by the use of the prolongation operator,

$$\boldsymbol{p}_f=\boldsymbol{P}\boldsymbol{p}_c. \tag{7}$$

In general, this will not solve (1) exactly no matter how accurate the coarse pressure is; all we can hope for is to compute a good and accurate approximation more efficiently than solving (1) directly on Ω.

3.1. Coarse system

To derive a linear system for $\boldsymbol{p}_c$ on the coarse grid, we insert the fine-scale approximation in (7) into (3) and apply the restriction operator,

$$\boldsymbol{R}\left(\boldsymbol{A}\left(\boldsymbol{P}\boldsymbol{p}_c\right)\right)=(\boldsymbol{RAP})\boldsymbol{p}_c=\boldsymbol{A}_c\boldsymbol{p}_c=\boldsymbol{Rq}=\boldsymbol{q}_c. \tag{8}$$

The physical interpretation of this system depends on the restriction operator used. Two variants are reported in the literature, either a control volume summation operator or a Galerkin operator, i.e.,

$$(\boldsymbol{R}_{cv})_{ji}=\begin{cases}1, & \text{if } \boldsymbol{x}_i\in\bar{\Omega}_j,\\ 0, & \text{otherwise,}\end{cases}\qquad\text{or}\qquad \boldsymbol{R}_G=\boldsymbol{P}^T.$$

Our focus is not on iterative performance, and hence it is natural to consider the control volume operator $\boldsymbol{R}_{cv}$ used in the classical MsFV method [8], which corresponds to setting the connection strength $(\boldsymbol{A}_c)_{ij}$ from coarse block $\bar{\Omega}_i$ into coarse block $\bar{\Omega}_j$ as the sum of the fluxes induced by the prolongation operator defined in block i across the interfaces of $\bar{\Omega}_j$,

$$\int\limits_{\partial\bar{\Omega}_j}\vec{v}_{P_i}\cdot\vec{n}\,dS\approx\sum_{(k,l)\in F_j}-T_{kl}(\boldsymbol{P}_{li}-\boldsymbol{P}_{ki})=\boldsymbol{A}_{ij}$$

where we have defined $\vec{v}_{P_i}$ as the velocity of the basis function of block i, F_j as the set of fine scale interfaces for $\bar{\Omega}_j$, represented as tuples of neighboring cells (k,l) and approximated the flux using (2).

The pressure obtained by prolongating the solution of (8) back to the fine scale is generally not an exact solution of (3), but we can easily compute fluxes that are conservative on the coarse grid since (8) imposes mass balance on this grid. To get fluxes that are conservative also on the subscale, we need to solve an additional local problem with the conservative, coarse-scale fluxes imposed as Neumann boundary conditions. Using these fluxes, it is possible to solve fine-scale transport to a high accuracy without the exact pressure being known. The disadvantage is that we risk producing negative coarse-scale transmissibilities, which may lead to unphysical solutions having non-monotone pressure values that violate the maximum principle, see e.g., [25].

To define a specific multiscale method, we must also describe in detail how to construct the prolongation operator P. As in most other multiscale methods, we construct P by piecing together a set of localized functions. However, before we can describe these so-called basis functions, we must provide more details about the coarse grid.

3.2. Coarse grids and support regions

Multiscale finite-volume formulations rely on a combination of a primal coarse grid and an auxiliary spatial characterization to obtain localized functions and assemble them together to form the prolongation operator. For the MsFV method, this additional mechanism is a dual coarse grid, whereas for the MsTPFA method [34], the additional mechanism is a set of localized partition-of-unity functions that are computed numerically over the fine grid. Previous research has shown that the choice of the primal grid (and the auxiliary spatial characterization) can have a pronounced impact on the solution quality. We therefore want a formulation that is as flexible as possible and enables basis functions to be computed for fairly general coarse grids. Even for problems posed on a Cartesian mesh without faults or pinched cells, the solution can be greatly improved with coarse grids that adapt to local features in the permeability field or in the flow patterns [25,34,43].

The primal coarse grid is defined through a partition vector that has a single unique indicator value per fine cell, so that cells with the same indicator value are agglomerated into coarse blocks. For structured grids, these indicators can simply be based on counting, forming logically hexahedral blocks on the coarse scale, whereas a wide variety of graph-based partitioning algorithms can be used in the unstructured case. Let F be the index set of fine cells and let C_j be the set of fine-scale indices corresponding to coarse block number j,

$$C_j \subseteq F, \qquad C_j \cap C_i = \emptyset \quad \forall \quad i \neq j, \qquad i, j \in [1, m], \qquad |F| = n.$$

Once we have defined a coarse grid, we must define the *support regions* that determine the support of the basis functions. If we let I_j denote the set of all points contained in the support region for coarse block $\bar{\Omega}_j$ and P_j the basis function of coarse block j, this implies that

$$P_j(\boldsymbol{x}) > 0, \quad \boldsymbol{x} \in I_j \qquad P_j(\boldsymbol{x}) = 0 \text{ otherwise.}$$

Likewise, we define the *support boundary* B_j as the index set of cells that are topological neighbors to the support region I_j, but are not themselves contained in it. We also define the *center* of a coarse block $\bar{\Omega}_j$ as $\boldsymbol{x}^c_j$. This is a single point, which in many cases *may* coincide with the centroid of the coarse block. For convenience, we also define the global support boundary G as the index set of all fine cells that are part of the support boundary of one or more coarse blocks,

$$G = B_1 \cup B_2 \cup \ldots \cup B_{m-1} \cup B_m.$$

Finally, for each fine cell that is member of the global support boundary, we define H_i to be the set of indices of the support regions the cell belongs to,

$$H_i = \{j \,|\, i \in I_j, i \in G\}.$$

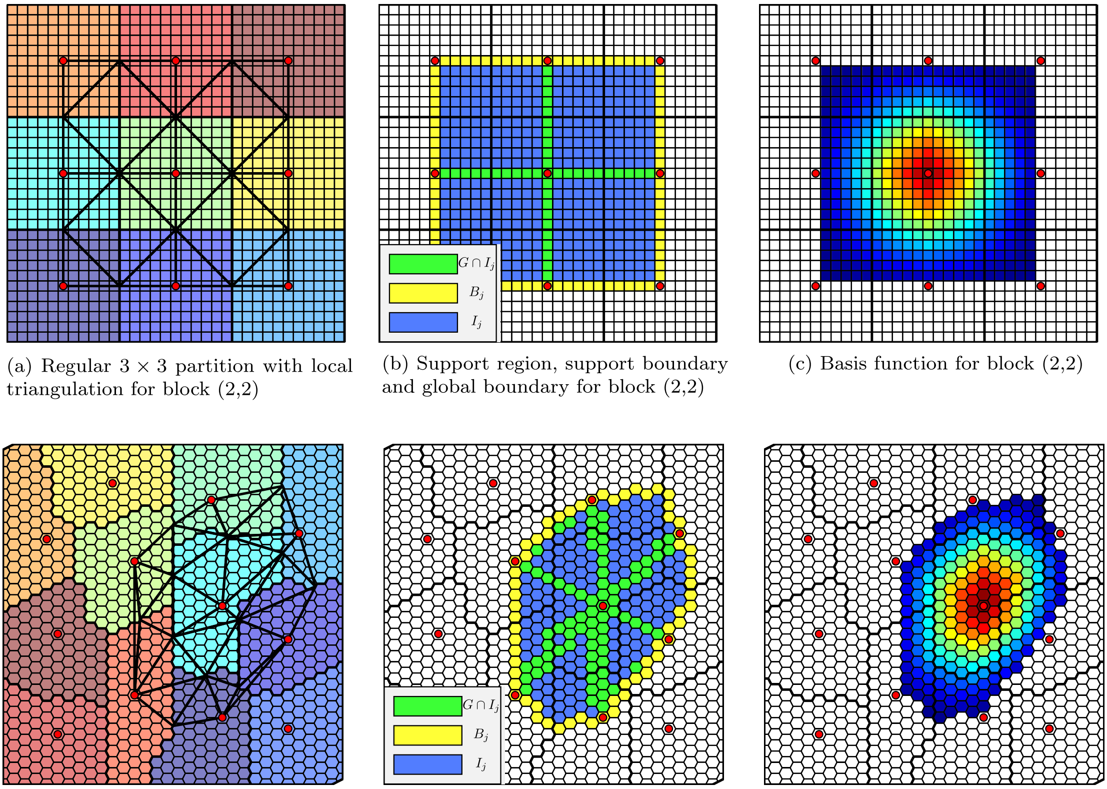
To make this notation easier to visualize, we refer to Fig. 1, which shows a regular partition for a uniform Cartesian grid and a semi-structured partition for a PEBI grid. The figure also shows the construction of support regions, support boundaries, and basis functions.

Careful numerical experiments have shown that rather than setting the block centroid as the block center, we should choose the block centroid as the fine cell whose centroid is closest to the geometric median of the fine-scale faces that bound the block. For regular coarse grids, the two choices coincide, but the geometric median gives basis functions of better quality for coarse partitions with large variation in block sizes and shapes. Fig. 2 shows an exaggerated example with a large block neighboring ten small blocks that each consists of a single fine-scale cell. In Fig. 2(b) the block center is chosen as the block centroid. Because of the disparity in size between the block and its neighbors to the west, the support regions are defined so that there is a relatively large region inside the block where only the basis function associated with the block itself has support. To ensure partition of unity, the basis function will therefore be constant and equal the maximum value of one in this region. Using the geometric mean instead, as in Fig. 2(c), means that the basis functions in the north, south, and east neighbors are supported in a larger portion of the center block, which reduces the constant region and improves the approximation quality of the prolongation operator significantly, so that it, for instance, can better reproduce a linear pressure drop.

To define the support region of block number i, we select all blocks that share a coarse node with $\bar{\Omega}_i$ and create a local triangulation based on the block centers and the centroids of all coarse faces that are shared by any two of these blocks, see Figs. 1(a) and 1(d). The support region is then defined as all cells within the selected coarse blocks whose centroids lie within the triangulation, and the support boundary is defined as all cells that share at least one face with cells in the support region. Support regions are not allowed to include any center cells aside from their own. Because all these relations can be produced using only topology information, block centers, and face centroids, the implementation is the same regardless of whether we identify support regions in a two-dimensional Cartesian grid or in a complex 3D unstructured grid. If needed, one can include a simple post-processing to ensure that each support region only consists of cells that are connected in the graph defined by the fine-cell faces.

3.3. Construction of basis functions

Most multiscale methods rely on numerical solution of localized flow problems to produce the basis functions that form the prolongation operator. These local problems are typically defined as some subset of the global problem with alternate



(a) Regular 3 × 3 partition with local triangulation for block (2,2)

(b) Support region, support boundary and global boundary for block (2,2)

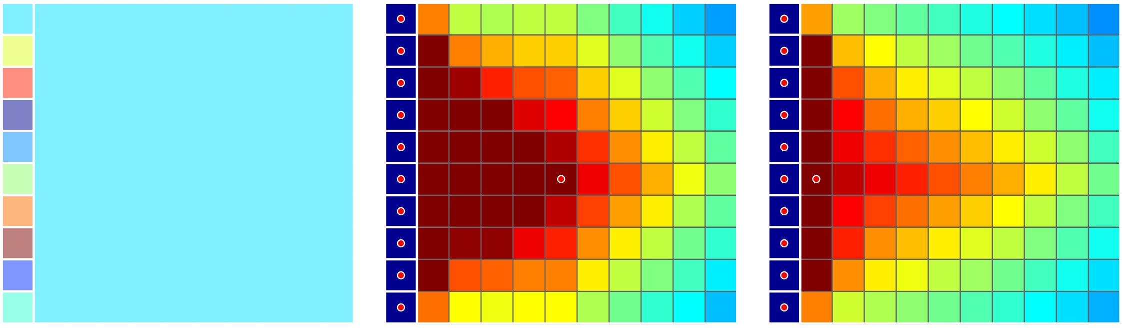
(c) Basis function for block (2,2)

(d) Semi-structured 4 × 3 partition and local triangulation for block (3,2)

(e) Support region, support boundary and global boundary for block (3,2)

(f) Basis function for block (3,2)

Fig. 1. Primal coarse grid and construction of a local support region with an associated basis function for a uniform Cartesian grid in the top row and a perpendicular bisector (PEBI) grid in the bottom row.



(a) Coarse block with refined neighbors

(b) Basis function using centroids

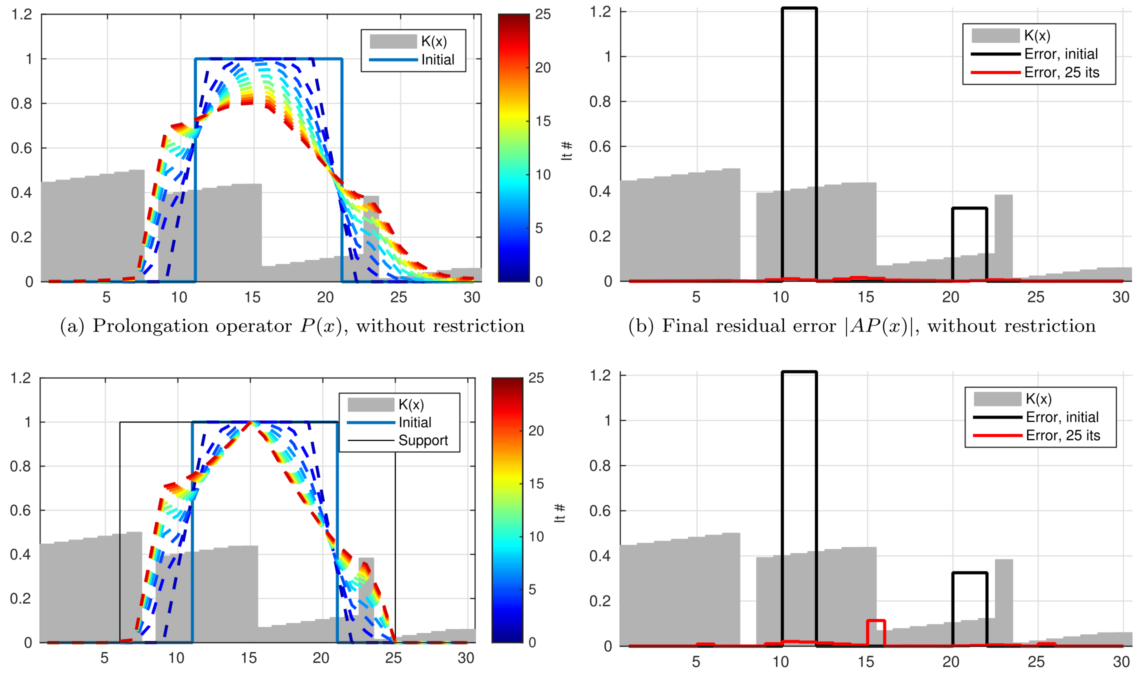
(c) Basis function using geometric mean

Fig. 2. The choice made for the definition of the center points affects the quality of the prolongation operator. This example shows a coarse block where the neighboring blocks to the west consist of a single cell, while the neighbors to the north, south, and east are of the same size as the block itself. The prolongation operator takes values in the interval [0, 1] and assumes a more correct, skewed hat shape when the block center is defined as the geometric median of the face centroids rather than as the block centroid.

boundary conditions imposed to capture the local features. Herein, we deviate from this and instead construct the basis functions using an iterative process. A similar approach can be found in some multigrid methods that employ a single step of a smoother applied directly to a simple prolongation operator to reduced local error (Jacobi interpolation), see [35].

The basis functions are initiated as the characteristic function of each coarse block,

$$P^0_{ij} = \begin{cases} 1 & \text{if } i \in C_j \\ 0 & \text{otherwise.} \end{cases}$$



(a) Prolongation operator $P(x)$, without restriction

(b) Final residual error $|AP(x)|$, without restriction

(c) Prolongation operator $P(x)$, with restriction

(d) Final residual error $|AP(x)|$, with restriction

Fig. 3. Illustration of how the iteration (9) and its restricted counterpart gradually smooth a basis function until it has very low residual error.

Nothing is preventing us from choosing some more intricate initial guess, but constant functions are convenient because they are trivial to construct and automatically provide partition of unity. We then define a local smoothing iteration,

$$P_j^{n+1} = P_j^n - \omega D^{-1} A P_j^n, \tag{9}$$

where D is a diagonal matrix that contains the diagonal entries of our (weakly) diagonally dominant system matrix A, and $\omega \in (0, 1]$ is a relaxation factor which we set to 2/3 for all cases considered in this paper. The value 2/3 is the optimal choice for Jacobi's method applied to Poisson's equation with constant coefficients. Better choices of ω will speed up convergence of the basis construction, but are not necessarily obvious for general problems. By iterating on the prolongation operator, we seek to make it algebraically smooth, i.e., reduce $\|AP\|_1$ as much as possible. This means that the residual error in the prolongation operator should be relatively smooth. Fig. 3 shows a simple 1D example of how the operator and the associated error change as more iterations are applied. Because each iteration modifies cell values based on the topological neighbors, we can see that the support of the basis functions will eventually grow to cover the entire domain. To avoid this, we use our already defined support regions and support boundaries to localize the updates.

Roughly speaking, the above construction (9) determines an increment for each basis function based on the local error and modifies this increment to limit the support to be within the support regions. This update is also used to determine convergence of the basis construction procedure:

1. Apply the smoother D to find the increment of the discrete basis function,
$$\widehat{\mathbf{d}}_j = -\omega D^{-1} A P_j^n.$$
2. Modify the update to avoid stencil growth outside of the support region and preserve partition of unity,
$$d_{ij} = \begin{cases} \dfrac{\widehat{d}_{ij} - P_{ij}^n \sum_{k \in H_i} \widehat{d}_{ik}}{1 + \sum_{k \in H_i} \widehat{d}_{ik}}, & i \in I_j, i \in G, \\ \widehat{d}_{ij}, & i \in I_j, i \notin G, \\ 0, & i \notin I_j. \end{cases}$$
3. Update basis functions
$$P_{ij}^{n+1} = P_{ij}^n + d_{ij}$$

4. Define local error outside of boundary regions,

$$e_j = \max_i(|\widehat{d}_{ij}|), \qquad i \notin G$$

5. If $\|\mathbf{e}\|_\infty >$ tol, go to Step 1, otherwise set $P = P^{n+1}$.

In practice, checking for convergence should only be done every tenth iterations or so, since a single iteration has negligible cost. We note that Step 1 of this process is well suited to parallel processing. As each value only depends on the fine neighbors, it can easily be computed using on streaming processors such as GPUs. The modifications in Step 2 at the global boundary depend on the values in several coarse blocks, so each basis function will depend on the previous value of the other basis functions with support in the same cell. While this does make the basis functions use information about each other, the dependence will only extend to the nearest neighbors during each step, and only in a subset of the cells, analogous to the matching boundary conditions used on edges in the classical MsFV.

Because we have explicitly enforced partition of unity in cells belonging to the global boundary, we must now show that the updates $\widehat{d}_{ij}$ preserve the same property in cells not on the boundary. We assume that the row sum of the matrix used for the iterations is zero and that the initial prolongation operator has partition of unity,

$$\sum_j A_{ij} = 0, \qquad \sum_j P^0_{ij} = 1 \quad \forall \quad i.$$

If needed, one can easily ensure that the iteration matrix has zero row sum by adjusting the diagonal elements. This requirement also applies to the classical MsFV method, so the same adjustment is used if special basis functions for wells, boundary conditions, and compressibility are not constructed. We can then write out the explicit update for a single cell i, summed over all coarse blocks,

$$\begin{aligned}\sum_j P^{n+1}_{ij} &= \sum_j P^n_{ij} - \frac{\omega}{A_{ii}} \sum_j \sum_k A_{ik} P^n_{kj} \\ &= 1 - \frac{\omega}{A_{ii}} \sum_k A_{ik} \left(\sum_j P^n_{kj} \right) = 1 - \frac{\omega}{A_{ii}} \sum_k A_{ik} = 1,\end{aligned}$$

showing that the update always preserves partition of unity.

For completeness, we will also verify that the proposed updates for cells in G satisfy partition of unity. If we recall that $\sum_{j\in H_i} P^n_{ij} = 1$ by assumption and that P^n_{ij} is nonzero only in H_i, we can explicitly write out the sum over basis functions for a cell i in G at step $n+1$,

$$\begin{aligned}\sum_{j\in\{1,\ldots,m\}} P^{n+1}_{ij} &= \sum_{j\in H_i} \left(P^n_{ij} + \frac{\widehat{d}_{ij} - P^n_{ij} \sum_{k\in H_i} \widehat{d}_{ik}}{1 + \sum_{k\in H_i} \widehat{d}_{ik}} \right) = 1 + \sum_{j\in H_i} \frac{\widehat{d}_{ij} - P^n_{ij} \sum_{k\in H_i} \widehat{d}_{ik}}{1 + \sum_{k\in H_i} \widehat{d}_{ik}} \\ &= 1 + \frac{\sum_{k\in H_i} \widehat{d}_{ik}}{1 + \sum_{k\in H_i} \widehat{d}_{ik}} - \frac{\sum_{k\in H_i} \widehat{d}_{ik}}{1 + \sum_{k\in H_i} \widehat{d}_{ik}} \sum_{j\in H_i} P^n_{ij} = 1.\end{aligned}$$

Fig. 4 illustrates how the prolongation operator changes to adapt to structures in the underlying medium. For comparison, we have also included plots of the corresponding prolongation operators for the MsFV method [8]. Note that while the MsRSB method coincides with MsFV for the case with homogeneous permeability and to a certain extent the lognormal Tarbert layers, there are large differences for the problems with anisotropy and channelized permeability sampled from Upper Ness.

To accurately represent Dirichlet boundary conditions, we use a similar approach as shown for the center block in Fig. 2(c). That is, we move the block center to a fine cell that is adjacent to the boundary; preferably to the cell that lies closes to the centroid of the block face. This way, we ensure that the corresponding basis function decays smoothly out from the Dirichlet boundary. Using this approach, we can reproduce a linear pressure drop in a homogeneous domain. No other special treatment is required for no-flow boundaries.

3.4. *Iterative multiscale formulation*

Multiscale finite-volume methods have a link to multigrid methods in the sense that they can be used as two-level methods in combination with a *smoother* step that takes care of localized errors. This can be used for error control, to treat compressibility and nonlinear behavior, or to systematically drive the fine-scale residual towards zero. In other words, the MsRSB method can be used in three different ways: as a linear solver for the fine-scale system (see [43]); as an approximate solver that only reduces the fine-scale residual below a prescribed, relaxed tolerance and still guarantees a mass-conservative approximation, or as a one-step approximate solver that is mass-conservative on the fine scale, but has no guarantee on the size of the fine-scale residual.

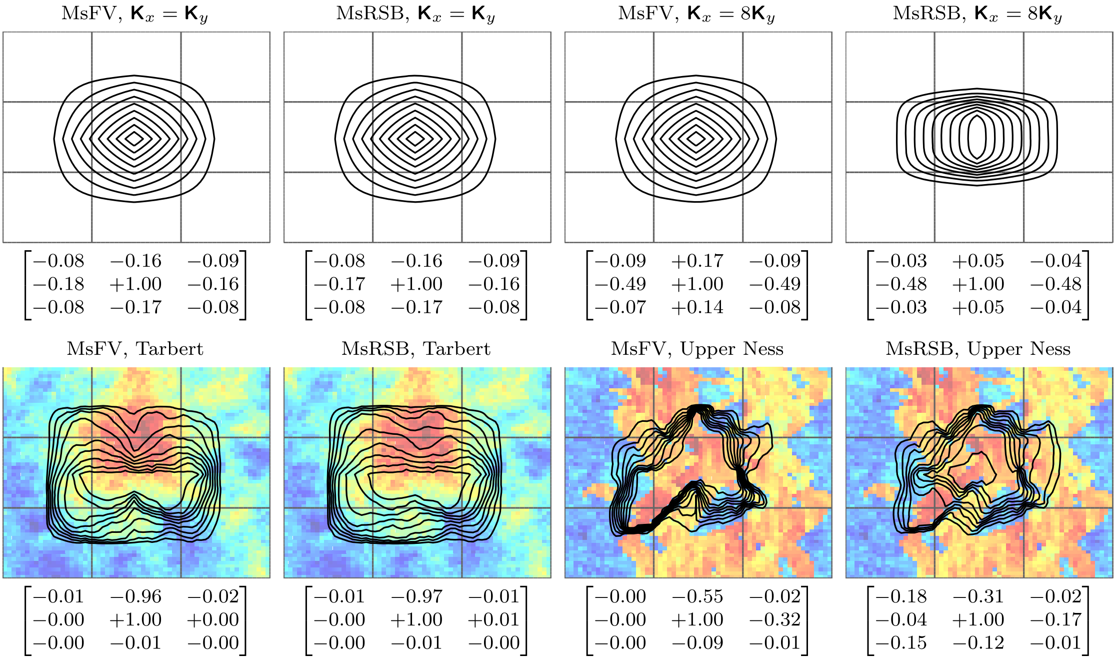


Fig. 4. Matrix-dependent interpolation operators for a single coarse block with different types of permeability types. The lognormal and the channelized permeabilities in the lower row are both sampled from the SPE10 dataset [45]. The matrices report the net fluxes into or out of the neighboring coarse blocks induced by a unit pressure differential.

Let P be the converged prolongation operator for a given coefficient matrix $\boldsymbol{A}$. To define an iterative scheme, we let the solution at step k be denoted $\boldsymbol{x}^k$ and introduce the defect

$$\boldsymbol{d}^k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k.$$

If we let $\boldsymbol{y}^k = S(\boldsymbol{d}^k)$ denote the smoother applied to the defect with initial guess zero, we can then write the next update as the previous solution with the smoothed update added in, along with a *coarse correction* that ensures that the update does not remove the coarse-scale conservative property of the solution,

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + P\big(A_c^{-1}R(\boldsymbol{d}^k - A\boldsymbol{y}^k)\big) + \boldsymbol{y}^k.$$

This iterative scheme relies on an inexpensive smoother for the updates. In the following, we use a incomplete LU-factorization with zero fill in (ILU0) for systematic iteration tests and a few Jacobi iterations for problems where we only want to reduce some local error. One pass of the smoother plus the coarse correction is termed a multiscale cycle. For the problems considered herein, we will let $\boldsymbol{x}^0 = 0$.

3.5. Flux reconstruction

The multiscale solution $\boldsymbol{p}_c$ is mass conservative on the coarse scale by construction. However, if we use the prolongated pressures to construct fine-scale fluxes from Darcy's law (see (2)),

$$v_{ij}^{ms} = -T_{ij}\big((\boldsymbol{P}\boldsymbol{p}_c)_i - (\boldsymbol{P}\boldsymbol{p}_c)_j\big), \tag{10}$$

these fine-scale fluxes will not be mass-conservative since $\sum_j v_{ij}^{ms} \neq q_i$. To get conservative fluxes on the fine grid (and hence also on any grid of intermediate resolution), we need to compute a new *reconstructed* pressure $\bar{p}$ to reconcile errors in the pressure gradient with the flux field as formulated for the classical MsFV method in [8]. We define the reconstructed pressure $\bar{p}$ by solving (1) locally for each coarse block C_i with flux boundary conditions obtained from the multiscale pressure over the coarse edges.

$$-\nabla \cdot (\mathbf{K}\nabla \bar{p}(\boldsymbol{x})) = q(\boldsymbol{x}), \boldsymbol{x} \in \bar{\Omega}_i \qquad \nabla \bar{p}(\boldsymbol{x}) \cdot \vec{n} = v^{ms} \text{ on } \partial\bar{\Omega}_i.$$

Once the reconstructed pressure is found, the velocity field inside each coarse block is found using Darcy's law. The fluxes over the coarse edges are the same as were used for boundary conditions. For the unstructured implementation, we use the operator form for posing these problems, see [46,16].

3.6. Treatment of wells

Herein, we consider wells that are either controlled by rate or by bottom-hole pressure. Wells perforated in a single fine cell are not much different from source terms or boundary conditions. For wells with multiple completions, potentially in different coarse blocks, we need to be a bit more careful. The standard way to treat a well is to use a set of source terms resulting from the pressure drop along the bore, i.e., if p_i is the bottom-hole pressure in the well and p_j the cell pressure, the well model gives us,

$$q_{ij} = \lambda_j^T W_I (p_j - p_i - \rho \vec{g} \nabla z),$$

where W_I is the productivity/injectivity index and λ^T the total mobility in the cells. In addition to this, a closure equation is defined per control: $\sum_j q_{ij} = \bar{q}_i$ for rate controls and $p_i = \bar{p}_i$ for pressure controls. In our multiscale framework we keep the control equations at the coarse scale to correctly account for inter-block flow on the coarse scale. For the bottom-hole controls, we take the well fluxes from the reconstructed pressure, as the pressure drop makes the reconstruction problems well posed and consistent with the outgoing block fluxes. For rate controlled wells, we use the fluxes defined by the prolongated pressure, as the well-to-cell flow is analogous to the flow between two coarse blocks.

4. Numerical experiments

The MsRSB method introduced above has been implemented using the Matlab Reservoir Simulation Toolbox (MRST), see [47–50], and is released as a part of the 2015a release. In the following we report the result of a series of numerical experiments we have run to validate the multiscale formulation, verify our implementation, and demonstrate the utility of the resulting solver. To this end, we consider a variety of test problems, from simple 2D Cartesian geometries to geological models representing petroleum reservoirs on the Norwegian Continental Shelf.

4.1. Spatial accuracy

To assess the spatial accuracy of the MsRSB method, we consider the single-phase model (1) applied to two different test cases: (i) the SPE 10 data set, which seem to be a de facto benchmark for new multiscale methods, and (ii) a model that uses the grid geometry and petrophysical properties from a simulation model of the Gullfaks field. For both models, we investigate the discrepancy between the multiscale approximation and the fine-scale reference solution measured by the scaled L^∞ and L^2 norms,

$$\|\boldsymbol{p}^{fs} - \boldsymbol{p}^{ms}\|_\infty = \frac{\max_{i\in F} |\boldsymbol{p}_i^{fs} - \boldsymbol{p}_i^{ms}|}{\max_{i\in F} |\boldsymbol{p}_i^{fs}|}, \qquad \|\boldsymbol{p}^{fs} - \boldsymbol{p}^{ms}\|_2 = \sqrt{\frac{\sum_{i\in F} |\boldsymbol{p}_i^{fs} - \boldsymbol{p}_i^{ms}|^2\,|\Omega_i|}{\sum_{i\in F} |\boldsymbol{p}_i^{fs}|^2\,|\Omega_i|}}, \tag{11}$$

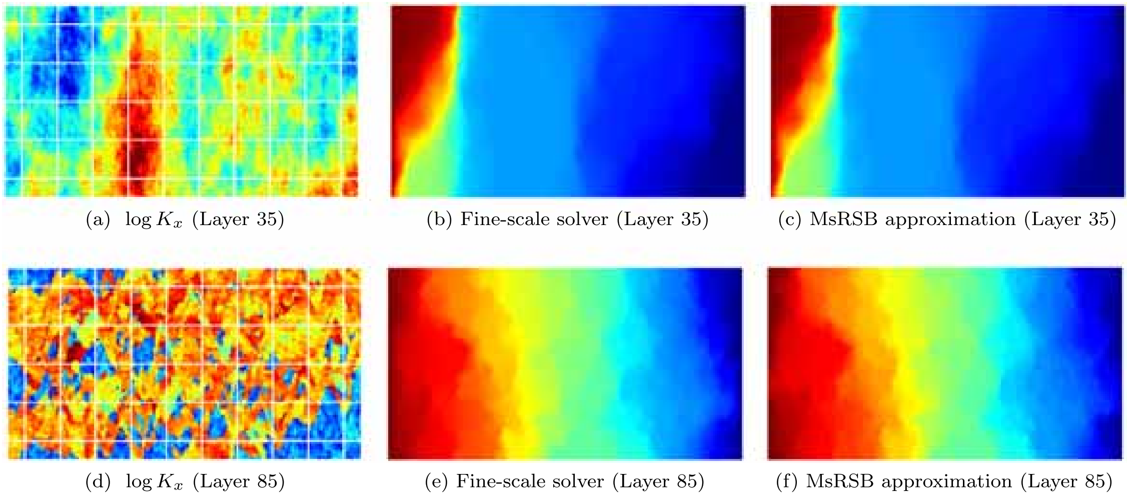
where $\boldsymbol{p}_i^{fs}$ and $\boldsymbol{p}_i^{ms}$ denote the pressure values computed in cell Ω_i by the fine-scale and the multiscale methods, respectively. Discrepancies in reconstructed fluxes are defined analogously.

4.1.1. SPE 10 data set

Model 2 from the 10th SPE Comparative Solution Project [45] was originally designed as a challenging benchmark for upscaling methods. The model is described on a 60 × 220 × 85 Cartesian grid with cells of uniform size 20 × 10 × 2 ft^3. The reservoir contains two sands sampled from a Brent sequence with very different heterogeneity. In the Tarbert formation found in the top 35 layers, the permeability follows a lognormal distribution, giving smoothly varying heterogeneities that are resolved quite well by most multiscale methods. The Upper Ness formation found in the bottom 50 layers is fluvial and consists of an intertwined pattern of long and high-permeable sand channels interbedded with low-permeable mudstone. The combination of very long correlation lengths and many orders-of-magnitude difference in permeabilities between neighboring cells makes Upper Ness very challenging to resolve accurately.

Horizontal layers First, we consider flow in two horizontal 60 × 220 slices with isotropic permeabilities sampled from the top and bottom layers of the model, subject to fixed pressure of one hundred bar on the left and zero bar on the right boundary. The domain is partitioned into coarse blocks made up of 10 × 20 fine cells so that the coarse grid blocks are square in the interior of the domain. Near the edges of the domain, we add coarse blocks that are half as wide as the other blocks in the x or y direction, respectively, see Fig. 5. In each of these blocks, we move the block center to the fine cell that lies closest to the face centroid of the block. This gives a total of 6 × 11 coarse blocks, which corresponds to an upscaling factor of 200 in the interior and 40 near the boundary.

Fig. 5 shows the permeability for both layers and compares the pressure fields computed by the fine-scale solver and by MsRSB using a single multiscale solve without subsequent iteration cycles. Table 1 reports the corresponding discrepancies measured in the relative L^2 and L^∞ norms defined in (11). For comparison, we also report discrepancies for the original MsFV method as implemented in the `msfvm` module of MRST, see [25] for details. Whereas the solution quality is generally very good for both solvers on the Tarbert subsample, MsRSB clearly outperforms MsFV on Upper Ness. Fig. 6 reports

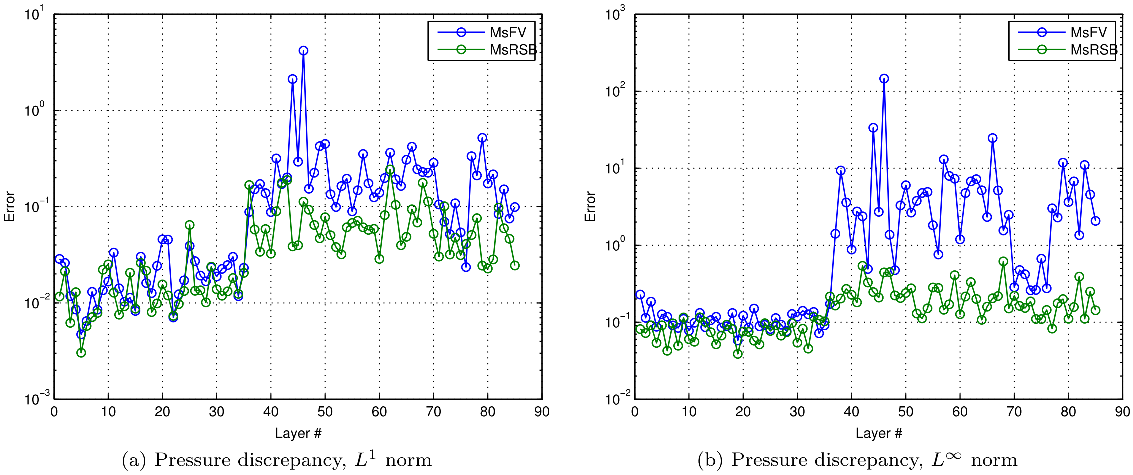

(a) $\log K_x$ (Layer 35) (b) Fine-scale solver (Layer 35) (c) MsRSB approximation (Layer 35)

(d) $\log K_x$ (Layer 85) (e) Fine-scale solver (Layer 85) (f) MsRSB approximation (Layer 85)

Fig. 5. Permeability and pressure solutions for the top and bottom layers of the second SPE 10 dataset. Flow is driven by a difference in the fixed pressures specified at the left and right boundaries.

Table 1
Discrepancy between the fine-scale solution and approximate solutions computed by the MsFV and MsRSB methods for the 2D test problems shown in Fig. 5. Total flux is the flux over the outflow edge normalized by the corresponding flux in the fine-scale reference solution.

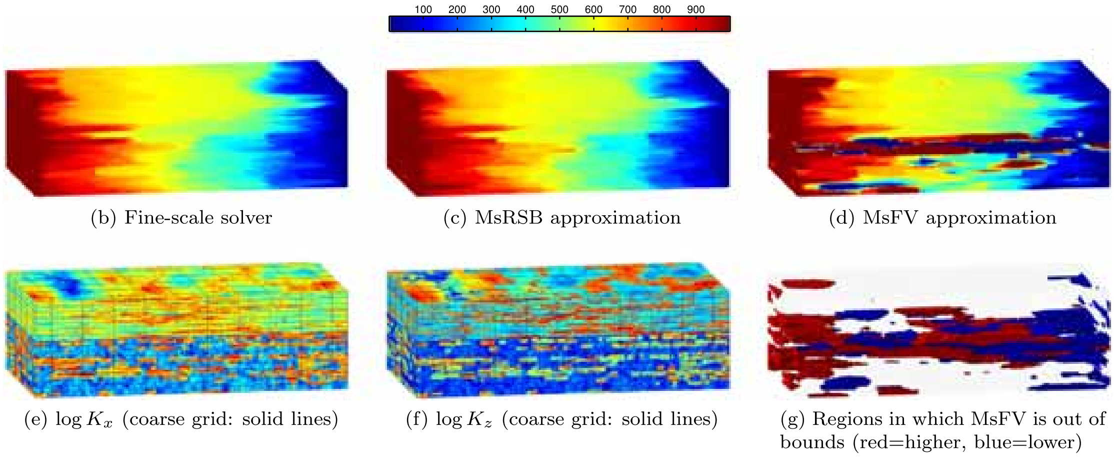
Setup of simulation			Pressure		Flux		
Formation	Layer	Solver	L^2	L^∞	L^2	L^∞	Total
Tarbert	35	MsFV	0.0313	0.0910	0.1138	0.4151	0.9696
		MsRSB	0.0204	0.0766	0.0880	0.4071	1.0121
Upper Ness	85	MsFV	0.2299	2.0725	0.4913	0.7124	0.8087
		MsRSB	0.0232	0.0801	0.1658	0.3240	1.0936



(a) Pressure discrepancy, L^1 norm (b) Pressure discrepancy, L^∞ norm

Fig. 6. Discrepancy between the fine-scale solution and approximate solution computed by the MsFV and MsRSB methods for all horizontal layers of the SPE 10 model.

discrepancies for similar experiments performed on all horizontal layers in the model. Several authors have independently shown that the MsFV method has issues with coarse-scale stability in the presence of channelized, high-contrast formations and will suffer from strong unphysical oscillations that may prevent iterative versions of the method from converging properly, see e.g., [51,21,25] and references therein. MsRSB is much more robust and does not suffer from such problems and therefore has approximately the same level of accuracy for the smooth and the channelized layers.



(b) Fine-scale solver

(c) MsRSB approximation

(d) MsFV approximation

(e) log K_x (coarse grid: solid lines)

(f) log K_z (coarse grid: solid lines)

(g) Regions in which MsFV is out of bounds (red=higher, blue=lower)

Fig. 7. Comparison of pressure solutions computed by the fine-scale solver and the MsFV and MsRSB methods on the full SPE 10 model using a 6 × 11 × 17 coarse grid. Flow is driven by a difference in fixed pressures specified at the west and east boundaries.

Table 2
Discrepancy between the fine-scale solution and approximate solutions computed by the MsFV and the MsRSB methods on two different coarse grids for the full SPE 10 model, see Fig. 7. Total flux is the flux over the outflow face normalized by the corresponding flux in the fine-scale reference solution.

Solver	Grid	Pressure		Flux		
		L^2	L^∞	L^2	L^∞	Total
MsFV	6 × 11 × 17	3.580	128.461	2.288	11.957	0.9110
MsRSB	6 × 11 × 17	0.039	0.309	0.397	0.487	1.2214
MsFV	P-adapted	0.054	0.259	1.418	1.397	1.8434
MsRSB	P-adapted	0.036	0.209	0.426	0.453	1.2179

Full 3D model The flow patterns in the 3D model are more complex, and the combination of strong anisotropy and higher aspect ratios poses additional challenges for multiscale methods. To coarsen the 1.1 million-cell fine-scale model, we use the same strategy as in the previous example with 10 × 20 × 5 fine cells per coarse block, giving coarse blocks of size 200 × 200 × 10 ft^3 in the interior of the domain. Fixed pressures of one hundred and zero bar are prescribed on the east and west boundaries, respectively.

Pressure solutions and discrepancies are reported in Fig. 7 and Table 2 and are in line with what we observed in 2D: Whereas MsFV and MsRSB both perform reasonably well in the upper part of the model, the original MsFV method becomes unstable in the lower channelized formation. For MsRSB, it is difficult to distinguish qualitative differences from the reference solution in Fig. 7, which is also confirmed by the quantitative comparison in Table 2. SPE 10 is a challenging benchmark in terms of heterogeneity, and the good accuracy obtained with the MsRSB method without any kind of grid adaption or smoothing iterations is quite remarkable.

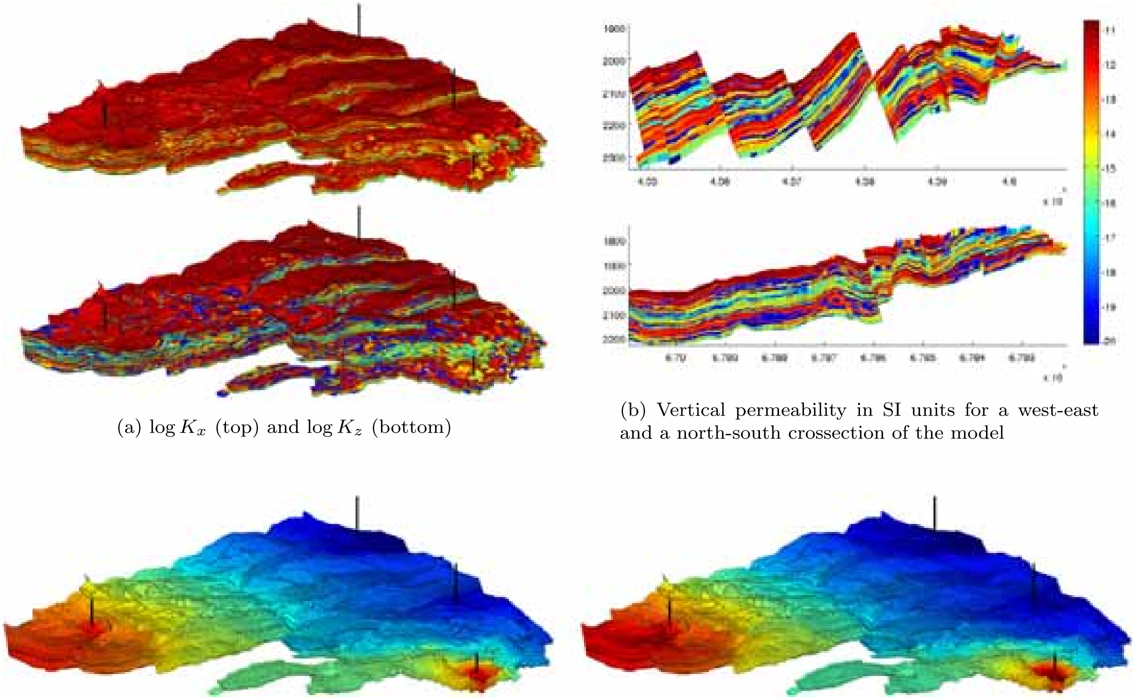
To mitigate the unstable behavior of the MsFV method, we can modify the coarse grid so that the control volumes adapt to the local structures in the prolongation operator. The adapted grid is defined by computing a new index set

$$\hat{C}_k = \{i \,|k = \operatorname{argmax}_j P_{ij}\}. \tag{12}$$

In other words, for each cell i we find the local prolongation operator that has the largest nonzero cell value and assign cell i to the corresponding coarse block. This gives a nonuniform grid that can be used as control volumes when formulating the coarse system. Table 2 shows that this gives a significant reduction in the discrepancy for the MsFV method, in particular in the L^∞ norm since the approximate solution now is kept within the bounds of the boundary conditions. The pressure discrepancy is also slightly reduced for MsRSB, but the reconstructed flux is significantly less accurate because of the local irregularity of the coarse blocks.

4.1.2. Gullfaks field model

Gullfaks is an oil and gas field located in the Norwegian sector of the North Sea that produces primarily from Brent sands, i.e., the same type of sedimentary environments as seen in the SPE 10 model. Unlike SPE 10, the Gullfaks field has a very complex structure and contains a large number of sloping faults, with angles varying from 30 to 80 degrees and throws from zero and up to three hundred meters; see [52] for a discussion of the structural geology. The simulation model is represented on a 80 × 100 × 52 corner-point grid in which 216 334 cells are active. Almost 44% of the cells have non-neighboring connections, and when the corner-point grid is turned into a matching polyhedral grid, the number of cell



(a) $\log K_x$ (top) and $\log K_z$ (bottom)

(b) Vertical permeability in SI units for a west-east and a north-south crossection of the model

(c) Pressure computed by the fine-scale solver

(d) MsRSB approximation (637 blocks, Metis)

Fig. 8. Computation of incompressible, single-phase pressure distribution using the grid geometry and the petrophysical data from a simulation model of the Gullfaks field. Pressure is set to 500 bar in the two injectors and 250 bar in the two producers.

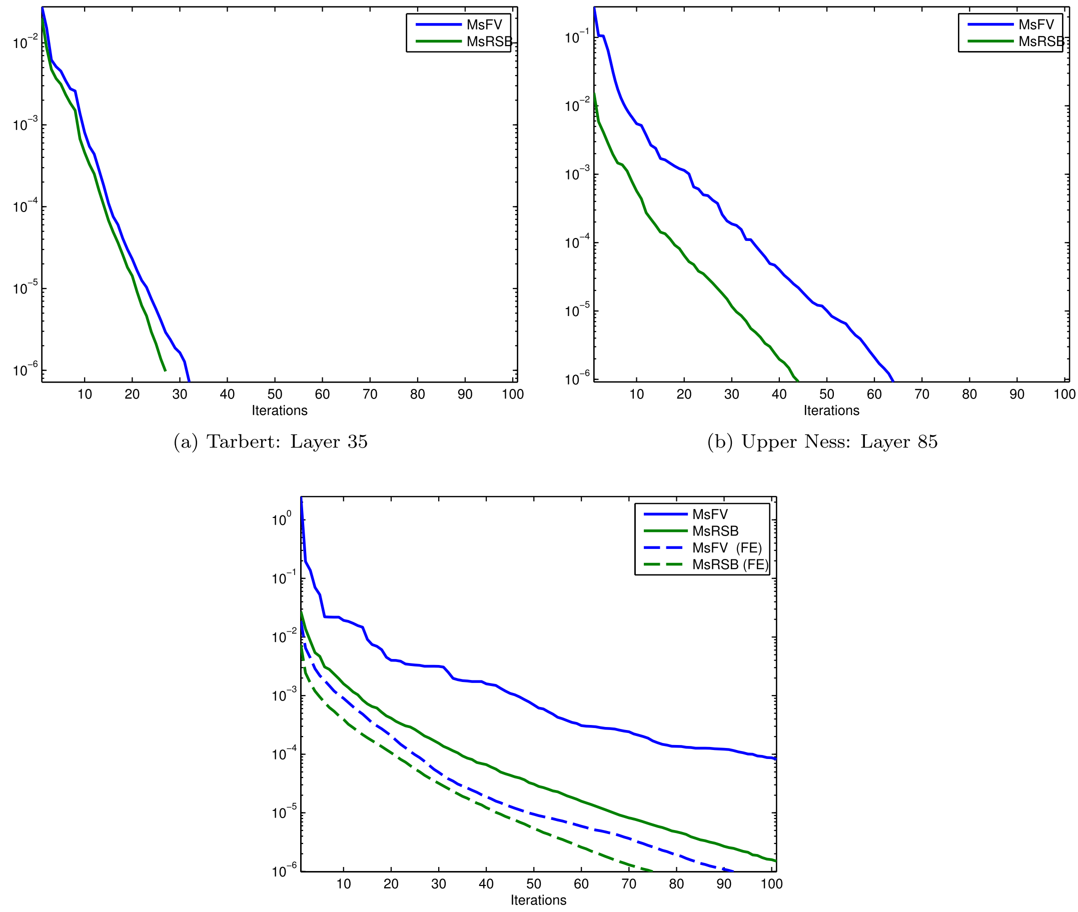
faces range from four to thirty-one. The combination of strong heterogeneity, large anisotropy and aspect ratios, degenerate cell geometries, and unstructured grid topology makes the Gullfaks model very challenging for any multiscale solver. Rather than considering the wells pattern that has been drilled in the actual formation, we create a significant pressure drop of 250 bar across the majority of the field using four wells, two producers and two injectors, placed quite arbitrarily near the perimeter and perforated through all layers of the model.

We consider three different partitions. The first is a coarse grid initially partitioned with $15 \times 15 \times 20$ fine cells per coarse block. The large coarsening factor in the vertical direction is chosen because of the many inactive layers and cells. Any coarse block intersected by a fault or divided into disconnected components by inactive cells is then split into two before coarse blocks with very few cells are merged into their neighbors, resulting in a semi-structured coarse grid with 416 blocks. The second coarse grid is constructed by the use of Metis [53] configured with the logarithm of the transmissibilities of the fine-scale system as weights for the edge-cut minimization algorithm and a target set to 416 blocks to match the number of blocks in the structured partition. We also consider another set of finer partitions constructed by the same methodology with approximately twice as many coarse blocks (1028 after processing) to demonstrate how the accuracy of the MsRSB method can be improved by coarse mesh refinement. We emphasize that no manual effort was required to create the three coarse partitions.

The approximate solution computed on the coarsest Metis-based grid can be seen in Fig. 8, while Table 3 reports the discrepancies from the fine-scale solution for all the three coarse grids. In all cases the approximate pressure values stayed inside the global bounds, i.e., in the interval from 250 to 500 bar. We note that the multiscale approximations are quite accurate and that refining the coarse partition improves the accuracy at the cost of a larger coarse system. The table also reports discrepancies after applying five multiscale cycles of ten Jacobi iterations each, which confirms that local errors are quickly removed by the inexpensive smoother. Altogether, the results are very promising in view of the combined challenge posed by the partially degenerate cell geometries, very complex grid topology, and the large permeability contrasts. Being able to handle models of this level of structural and stratigraphic complexity in a robust and automated fashion is essential if the goal is to bring multiscale methods closer to practical usage. The interested reader can also consult [43] for a discussion of MsRSB used as an iterative linear solver for a case with seven injection and eleven production wells.

Table 3
Discrepancy between the fine-scale solution and approximate MsRSB solutions computed for the Gullfaks model shown in Fig. 8. Total flux is the net flow out of the injectors normalized by the corresponding net flow from the fine-scale reference solution.

Solver	Iter.	Coarse grid		Pressure		Flux		
		Block type	DoF	L^2	L^∞	L^2	L^∞	Total
MsRSB	–	15 × 15 × 20	416	0.032	0.102	1.256	1.329	1.639
MsRSB	5	15 × 15 × 20	416	0.033	0.100	0.606	0.865	1.283
MsRSB	–	Metis	416	0.032	0.086	1.399	2.008	1.374
MsRSB	5	Metis	416	0.018	0.067	0.538	0.931	1.099
MsRSB	–	10 × 10 × 10	1028	0.028	0.597	2.165	4.143	1.502
MsRSB	5	10 × 10 × 10	1028	0.024	0.089	0.545	0.430	1.202
MsRSB	–	Metis	1028	0.015	0.112	1.347	1.399	1.310
MsRSB	5	Metis	1028	0.011	0.027	0.476	0.675	1.085



(a) Tarbert: Layer 35

(b) Upper Ness: Layer 85

(c) Full 3D model

Fig. 9. Convergence history for GMRES-MS with the MsFV or MsRSB prolongation operators on 2D subsets and the full SPE 10 model.

4.2. Multiscale methods as iterative solvers

The examples presented so far have only used multiscale methods as approximate solvers that are guaranteed to produce a conservative flux field regardless of the accuracy of the approximation. However, as explained in Section 3.4, multiscale methods can also be used as iterative solvers for the fine-scale system, e.g., in combination with GMRES. To demonstrate this capability for the MsRSB method, and compare its performance to that of the iterative MsFV method, we return to the SPE 10 test cases from Section 4.1.1. Fig. 9 reports the convergence of the two multiscale methods used as iterative solvers for two horizontal layer as well as on the full 3D model. On the smooth Tarbert layer, both methods are able to reduce

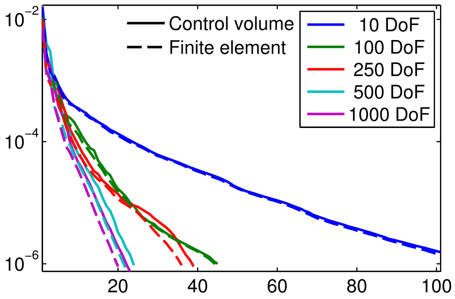


Fig. 10. Convergence history for GMRES-MS for the Gullfaks field model with different coarsening ratios.

the fine-scale residual six orders of magnitude in approximately thirty iterations (27 iterations for MsRSB and 32 for MsFV). On the channelized Upper Ness layer, both methods require more iterations, but because of the improved stability of the MsRSB prolongation operator, the residual for MsRSB is significantly lower than for MsFV and decays faster. For the full 3D model, the MsFV prolongation operator tends to produce solutions that are outside of the bounds, as shown in Fig. 7. Reduction of the residual is therefore mainly left to GMRES, which explains the slow convergence observed in Fig. 9(c). If the original mass-conservative control-volume restriction operator $\boldsymbol{R}_{cv}$ is replaced by the Galerkin operator $\boldsymbol{R}_G$, as suggested e.g., by [21], the MsFV method exhibits an acceptable convergence that is somewhat faster than MsRSB with finite-volume restriction and slightly slower than MsRSB with Galerkin restriction.

We also ran a set of tests on the Gullfaks model used in Section 4.1.2 where we varied the number of coarse blocks to estimate the impact coarsening has on convergence for an essentially unstructured model. We used Metis to produce coarse grids with 10, 200, 250, 500 and 1000 blocks. The convergence is shown in Fig. 10 where we see that adding more degrees of freedom appears to increase the convergence rate. Rapid convergence with a larger coarse system is to be expected, as more error modes will be in the null space of the coarse system. The choice of restriction operator does not have a large impact on the convergence rate, as the MsRSB operator produces a stable coarse system in either case.

Ref. [43] reports a more thorough assessment and comparison of the MsRSB, MsFV, and MsTPFA [34] prolongation operators used as pure multiscale solvers or as part of a GMRES iterative solver. The results confirm what we observed above: The MsFV method gives accurate multiscale solutions and converges rapidly on the smooth Tarbert formation irregardless of the restriction operator used. On Upper Ness, the method is accurate and efficient if one uses Galerkin reconstruction. The MsTPFA method is more robust but less accurate and efficient than MsFV on Cartesian partitions. However, if we use Metis to compute a coarse partition that adapts to contrasts in the fine-scale transmissibilities, the MsTPFA method becomes as accurate and efficient as MsFV (which generally cannot be applied to such adapted grids). In all tests, however, the MsRSB method performs equally well or better than the other two methods and is less affected by the choice of restriction operator and type of coarse partition.

4.3. Multiphase flow

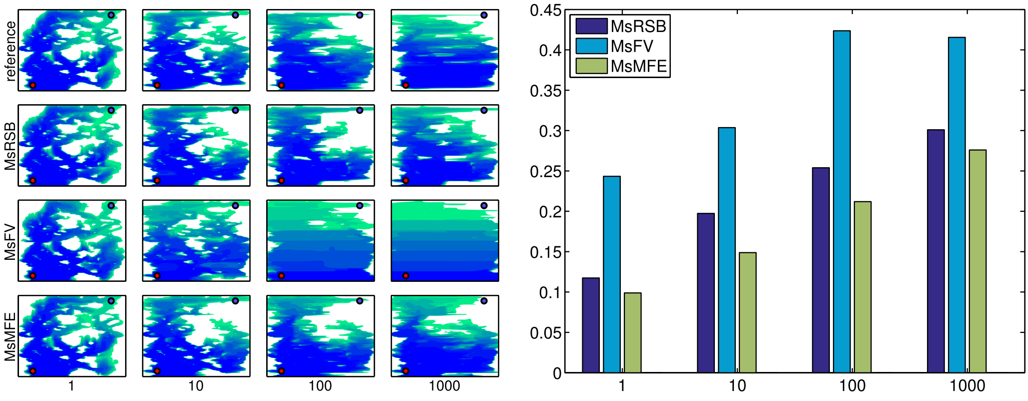
So far, we have mainly investigated how accurate the multiscale method is able to resolve the pressure in a given L-norm. In practical simulations, it is more important that the multiscale method gives fluxes that transport saturations/compositions correctly. For incompressible flow simulation, in particular, the pressure does not appear explicitly in the transport equation (6) and only influences the fluid displacement implicitly. Pressure fields are usually also much smoother than the associated flux fields and hence primarily reflects large-scale heterogeneous structures. To assess how well the multiscale methods resolve the influence of small-scale heterogeneous structures, we will investigate how the multiscale approximation affects the accuracy of the transport equation (6), measured using a relative L^1 norm

$$\|\phi S^{fs} - \phi S^{ms}\|_1 = \frac{\sum_{i\in F} \phi_i |S_i^{fs} - S_i^{ms}|\,|\Omega_i|}{\sum_{i\in F} \phi_i |S_i^{fs}|\,|\Omega_i|}. \tag{13}$$

Weighting the error by pore volumes makes this the error in spatial mass distribution for incompressible flow. To this end, we will, as in the previous sections, use a variety of test problems, from simple slices of the SPE 10 data set to a real field model.

4.3.1. Sensitivity to aspect/anisotropy ratios

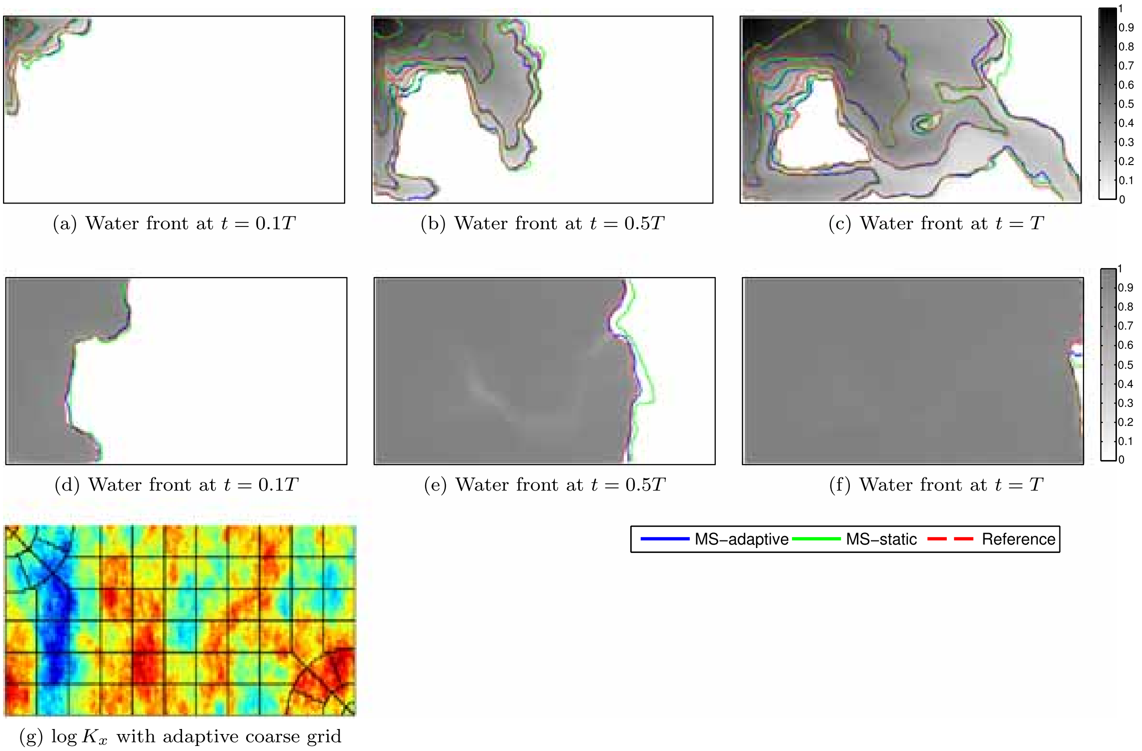
Grids used in field and sector models predominantly have much larger cells in the horizontal than in the vertical direction, or likewise have much smaller vertical than horizontal permeability. Hence, it is important that multiscale methods are robust with regards to large aspect and anisotropy ratios. It is well known that the classical MsFV method is sensitive to high anisotropy and large grid aspect ratios and may produce approximate solutions having strong localized circulations



(a) Saturation profiles

(b) Discrepancy in L^1 norm

Fig. 11. Aspect ratio test for solutions computed by the MsRSB, MsFV, and MsMFE methods on a uniform 4 × 8 coarse grid covering a 60 × 120 subset of Layer 85 of the SPE 10 data set that has been stretched a factor 1, 10, 100, and 1000 in the y-direction.


(a) Water front at $t = 0.1T$

(b) Water front at $t = 0.5T$

(c) Water front at $t = T$

(d) Water front at $t = 0.1T$

(e) Water front at $t = 0.5T$

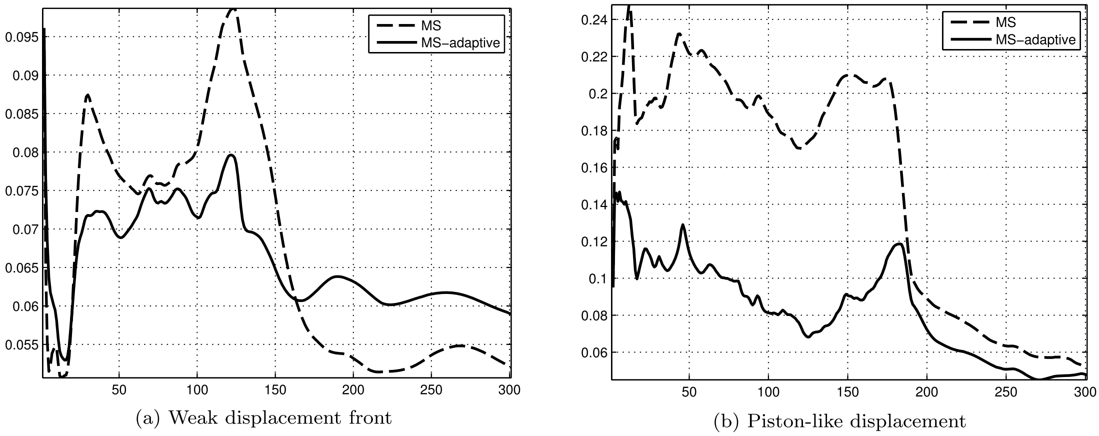
(f) Water front at $t = T$

(g) $\log K_x$ with adaptive coarse grid

Fig. 12. Saturation profiles for the two-dimensional waterflood example. The reference saturation field is shown in graytones, with contour lines of the two multiscale solutions superimposed. The first and second rows correspond to weak and piston-like displacement, respectively.

in the reconstructed fine-scale flux field, see e.g., [54]. This will, in turn, give smeared saturation fronts in which the subscale resolution is completely lost because of the strong circulations. In [55], it is shown that deficiencies appear even for homogeneous cases because of the bilinear nature of the basis functions. The problem can be mitigated to a certain degree by a careful choice of boundary conditions for localization of basis functions [56,55].

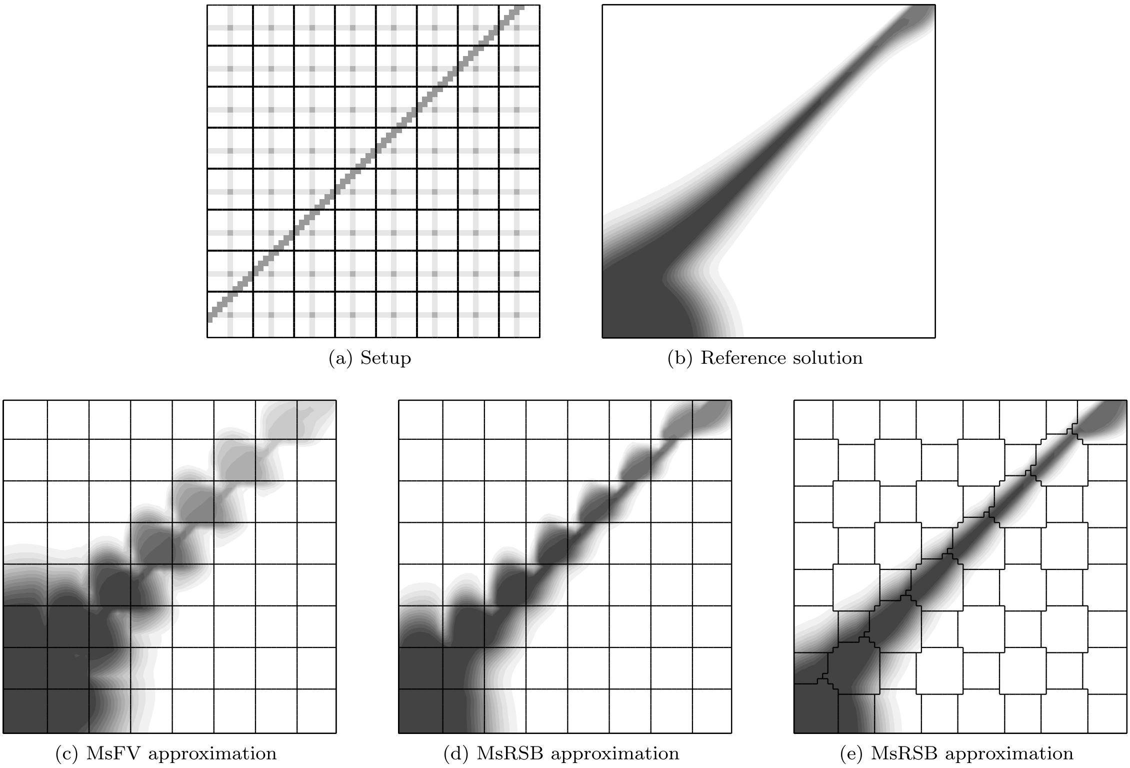
For the coefficients in the discretized pressure equation, stretching the grid is equivalent to increasing the anisotropy ratio. We consider a 60 × 120 subset of Layer 85 of the SPE 10 data set, giving a square domain with $L_x = L_y = 1200$ ft in which water is injected from a well operating at fixed rate near the south-east corner and fluid is produced from a well operating at constant pressure near the north-east corner. We will compare the MsRSB, MsFV, and MsMFE multiscale



(a) Weak displacement front

(b) Piston-like displacement

Fig. 13. Saturation error measured in the relative L^1 norm (13) as function of time step number for water injection in the top layer of the SPE 10 model.



(a) Setup

(b) Reference solution

(c) MsFV approximation

(d) MsRSB approximation

(e) MsRSB approximation

Fig. 14. Pathological example of a high-contrast medium. In the setup, dark color indicates the channel having 100 times higher permeability than the background field, solid lines show the coarse blocks, and light colors show cells along the edges of a dual coarse grid. The next four plots show saturation profiles after the injection of 0.25 pore volumes of water from a well placed at the center of the lower-left coarse block.

methods with the same uniform 4 × 8 coarse grid for all three methods. To assess how well the multiscale methods resolve the transport properties of the flux field, we simulate the injection of one half pore volume of water using linear relative permeabilities with unit mobility ratio (i.e., $\lambda_w(S) = S$ and $\lambda_n(S) = 1 - S$), for which the equation system (5)–(6) becomes fully decoupled and can be solved using a single pressure step.

Fig. 11 reports saturation profiles and discrepancy from the fine-scale solution on four different grids with $L_y = nL_x$, for $n = 1$, 10, 100, and 1000. The MsFV method has significantly larger errors than the other two methods and for aspect ratios 100 and 1000, all fine-scale details in the saturation field are lost. MsRSB is slightly less accurate than MsMFE, but although

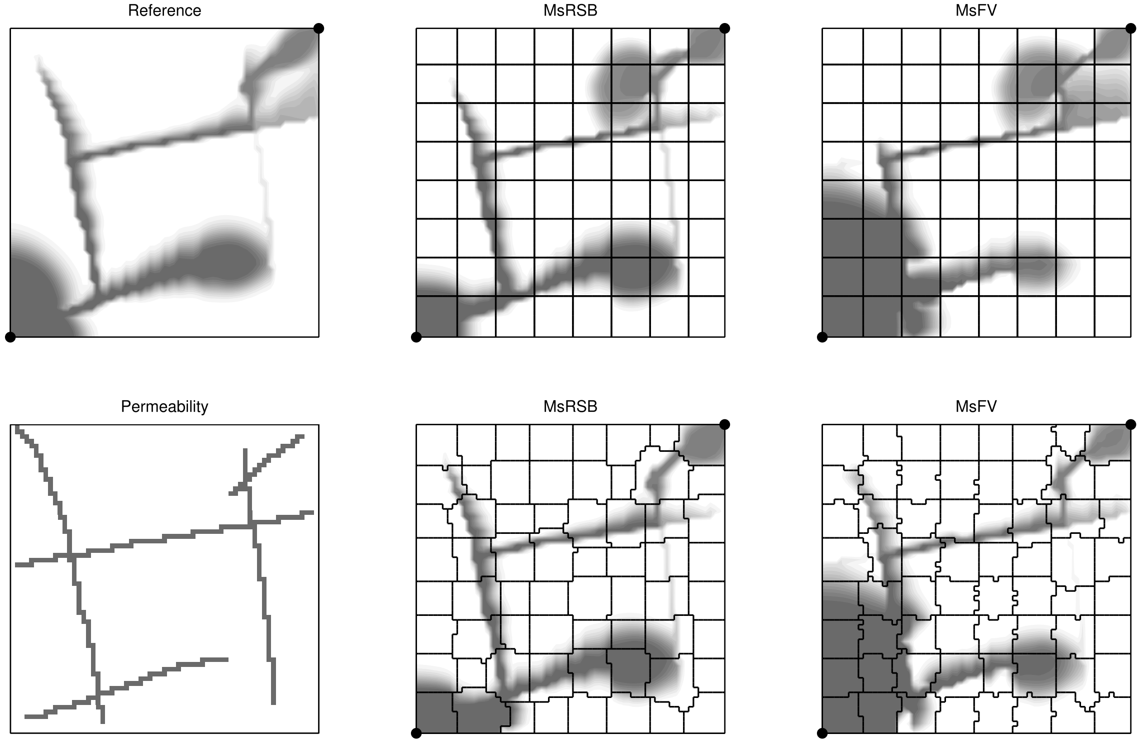


Fig. 15. Flow in a network of thin channels having one thousand times higher permeability than the surrounding rock. The plots show saturation profiles computed by the MsRSB and MsFV method on a Cartesian coarse grid and a coarse grid adapted to the basis functions.

both methods become gradually less accurate in a pointwise sense as the aspect ratio increases, they manage to maintain a reasonable prediction of the qualitative behavior of the flow pattern.

4.3.2. Temporal accuracy of multiscale approximation

There are two different ways of using a multiscale solver for time-dependent problems. The first approach is to keep the same basis functions throughout the whole simulation, and either hope that the iterations can efficiently account for mobility changes or accept the corresponding reduction in local accuracy if the multiscale method is used without iterations. Alternatively, one can update the basis to account for dynamic changes in mobility. The usual way to do this is to recompute basis functions locally whenever the total mobility changes significantly. The key to get enhanced efficiency when using multiscale methods is to perform as few subgrid computations as possible and only where it is necessary. Smoothed basis functions are particularly efficient in this respect since they do not need to be completely regenerated: Because the underlying iterative process can start from any function having partition of unity, we can simply restart the iteration process with changed mobilities and continue until the basis functions are sufficiently smooth again.

The dynamic character of an incompressible two-phase flow system is often quantified by the ratio of the end-point values of the total mobility, usually called the mobility ratio for brevity. To discuss this, we consider a special choice of relative mobilities

$$\lambda_w(S) = MS^2, \qquad \lambda_n(S) = (1-S)^2, \qquad 0 \le S \le 1. \tag{14}$$

In this case, the total mobility satisfies $\lambda(0) = 1$ and $\lambda(1) = M$, so that the end-point mobility ratio is M. Different end-point mobility ratios give rise to very different flow scenarios. With $M > 1$, we have an unstable displacement in which the injected fluid is more mobile than the resident fluid and will therefore tend to develop viscous fingers that penetrate rapidly through the less viscous resident fluid. For $M < 1$, on the other hand, the injected fluid is less viscous than the displaced fluid, giving a stable displacement characterized by a strong, piston-like displacement front. To correctly predict the speed of the displacement front, the multiscale method must accurately account for dynamic changes in the mobility. As noted by [54], resolving stable displacements is more challenging than resolving unstable displacements, for which the changes in the mobility field will be smooth and relatively small because of the weak displacement front. For a stable displacement, the propagation of the strong displacement front will induce large and abrupt changes in the total mobility field, so that it may deviate significantly from the mobility field used to compute the basis functions that make up the multiscale prolongation operator.

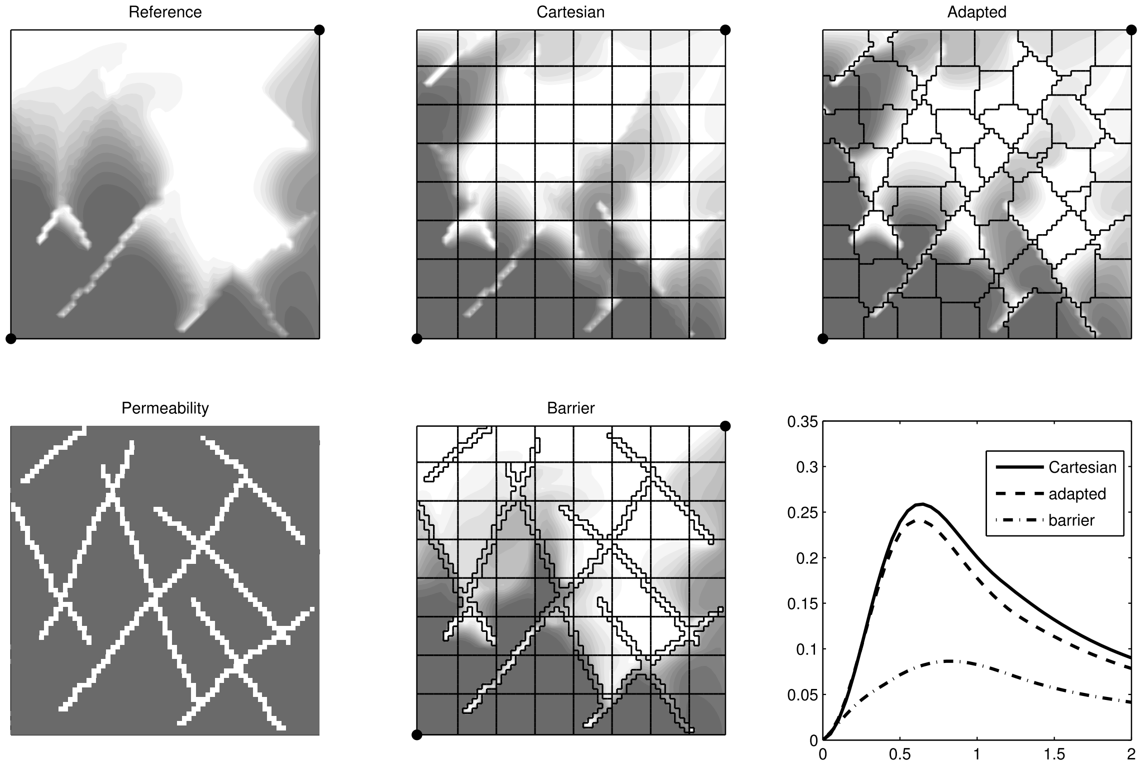


Fig. 16. Flow in a network of thin flow barriers having one thousand times lower permeability than the surrounding rock. The plots show saturation profiles after 0.5 PVI computed by the MsRSB method on a Cartesian coarse grid, on the same grid but with restriction operator adapted to the prolongation operator, and on a coarse grid that adapts to the barriers.

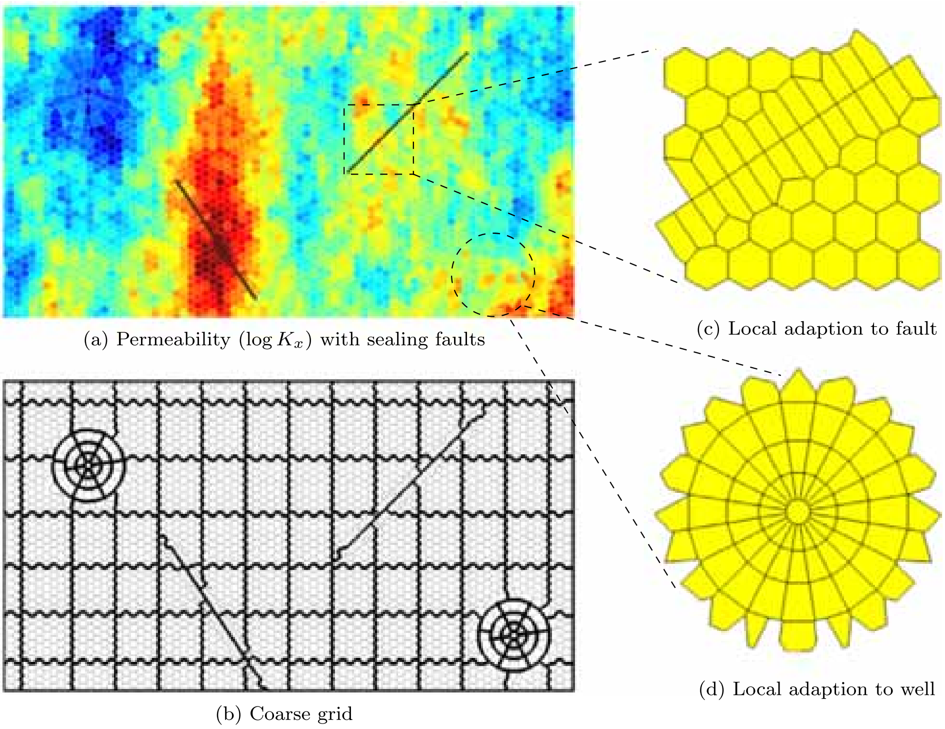
To compare the two multiscale approaches, updating or not updating basis functions, we consider the top layer of the SPE 10 model initially filled with oil and produced by a quarter five-spot well pattern with water injected in one corner and fluid produced in the diagonally opposite corner. The two wells are controlled by pressure, which gives a more challenging test than rate-based injection because the inflow and outflow now depends directly on the pressure distribution. Fig. 12 shows the saturation of the injected fluid for $M = 0.1$ and $M = 10$ at three different times in the interval $[0, T]$. To improve the prediction of pressure, we adapted the coarse grid by applying a radial refinement near the wells as shown in Fig. 12(g). For the weak displacement front there is good agreement between the multiscale and the reference solutions, even if basis functions are not updated throughout the simulation. For the sharp front, however, the multiscale method with static basis functions overestimates the speed of the leading shock. This error accumulates during the simulation and results in earlier breakthrough. A plot of the saturation error over time in Fig. 13 shows that the error is larger with static basis functions in both cases, but the difference in errors is more pronounced for the piston-like displacement.

4.3.3. High-contrast media

Extensive numerical experiments have shown that contemporary multiscale methods provide approximate solutions of good quality for highly heterogeneous media. However, cases with large permeability contrasts are generally challenging and it is not difficult to construct pathological test cases on which a particular method fails to produce accurate solutions.

Diagonal channel Ref. [54] proposed a simple and illuminating example consisting of a narrow high-permeable channel in a low-permeable background, where the channel is aligned with the diagonal direction of the grid as shown in Fig. 14(a). For the simulation, we use linear relative permeability, unit viscosity for both fluids, a single pressure step, and twenty time-steps in the implicit transport solver.

It is well known that high permeability contrasts along the edges of the dual grid can give poor localization for multiscale methods. If the channel intersects the faces of the primal coarse grid, the pressure extrapolation used to localize basis functions will start in a low-permeable region at a dual vertex, cross the high-permeable channel along the dual edge, before ending in a low-permeable region at the dual vertex on the opposite side of the channel. The corresponding basis function will overestimate flux between the high and low-permeable regions and cause a saturation front propagating to the high-permeable channel to leak out into the surrounding low-permeable background, as shown in Fig. 14, or in the


(a) Permeability ($\log K_x$) with sealing faults

(c) Local adaption to fault

(b) Coarse grid

(d) Local adaption to well

Fig. 17. Unstructured test case: Fine-scale permeability field, adapted coarse grid, and zoom of local refinements in the fine grid.

worst case, lead to non-monotone coarse-scale operators, e.g., as discussed in [25]. The same type of problem occurs for the MsMFE method, which relies on degrees of freedom associated with the faces of the coarse grid, if the channel is shifted slightly so that it intersects the vertices of the coarse blocks.

The problem can be mitigated by adapting the restriction operator to the local structures in the prolongation operator as discussed in Section 4.1; that is, by assigning cell i to block j if $P_{ij} \geq P_{ik}$ for all $k \neq j$. Fig. 14(e) shows that the saturation profile is significantly improved by using MsRSB on the resulting grid. Results for MsFV are almost identical and are not reported. Because the adaptive procedure relies on comparing floating point numbers, the grid is also perturbed away from the channel. This numerical artifact has no effect on the accuracy for this particular case and could easily have been removed if we had used a slightly more sophisticated implementation.

Multiple channels Our next example considers a slightly more complex case with a network of thin channels having one thousand times higher permeability than the surrounding rock. Fig. 15 reports fine-scale solutions and multiscale approximations computed with the same computational setup as for the diagonal channel, except for changes to the permeability field. Adapting the restriction operator clearly improves the qualitative prediction of the flow patterns for both multiscale methods. However, the approximation errors are quite large for both methods and if the purpose is to accurately predict production profiles or the evolution of the saturation field, extra iterations will be necessary. Plots of production profiles and saturation errors are not included for brevity, but show that MsRSB predicts well production much more accurately than MsFV and also gives much lower L^1 errors in the saturation fields. Interestingly, while adapting the restriction operator improves saturations errors significantly for MsRSB, the match in production profile becomes slightly worse. For MsFV, adapting the restriction operator improves both production profiles and the saturation field.

Our experience after having run a number of similar experiments of high-contrast media is that adapting the restriction operator is a simple precaution strategy to improve solution quality near high-permeable channels, at least when these channels are scattered relatively sparsely throughout the domain. Once exception is dense systems of narrow, high-permeable channels as found in fractured media, for which our experiments indicate that is better to describe the high-permeable fractures as lower-dimensional objects in the multiscale method, see [57]. This will be discussed in more detail in a forthcoming paper.

Multiple flow barriers Previous experience with the MsMFE method has shown that barriers are best represented in a multiscale solution if one adapts the coarse blocks to follow the barriers, see [58,59]. Fig. 16 shows a case containing a network of narrow barriers, simulated with the same setup as in the two previous examples, except that we now inject

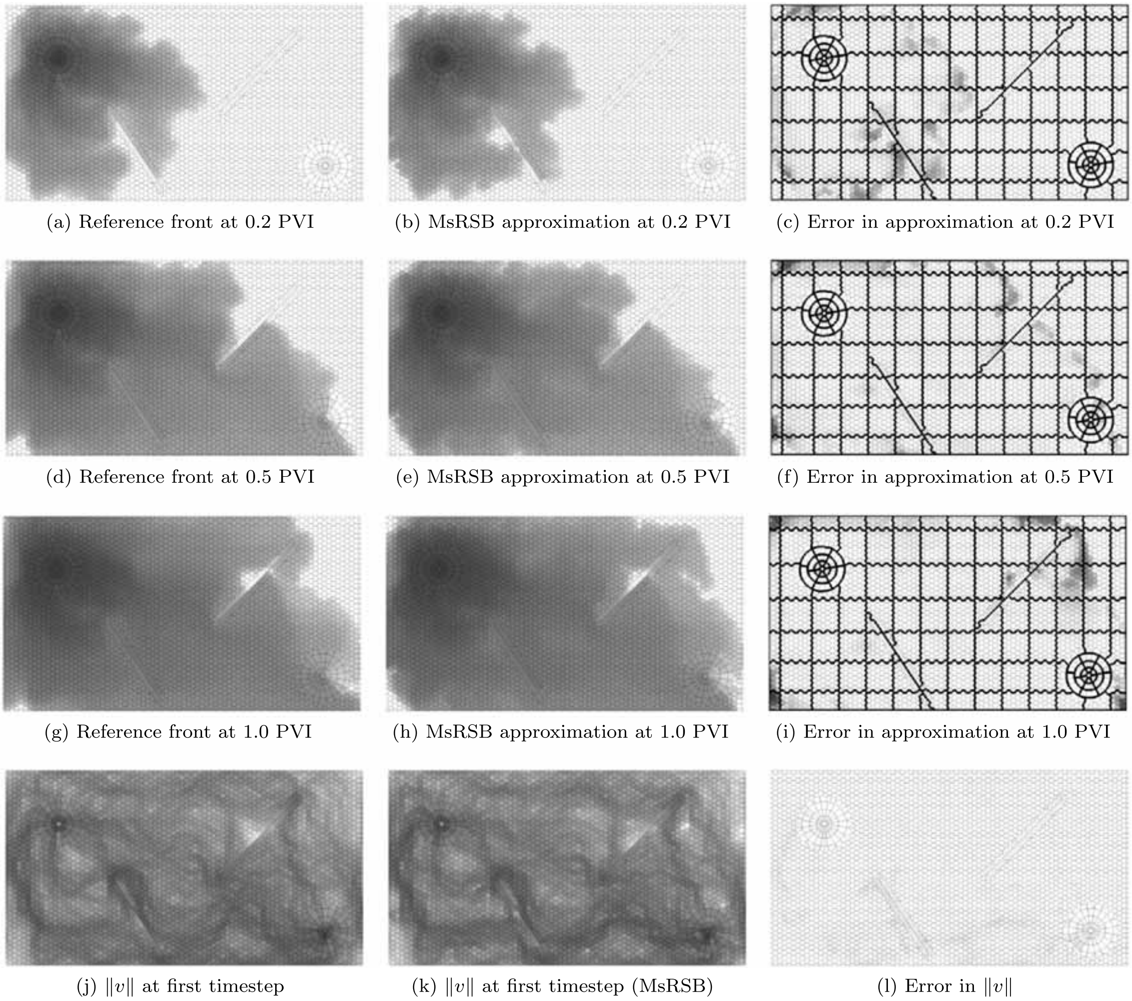


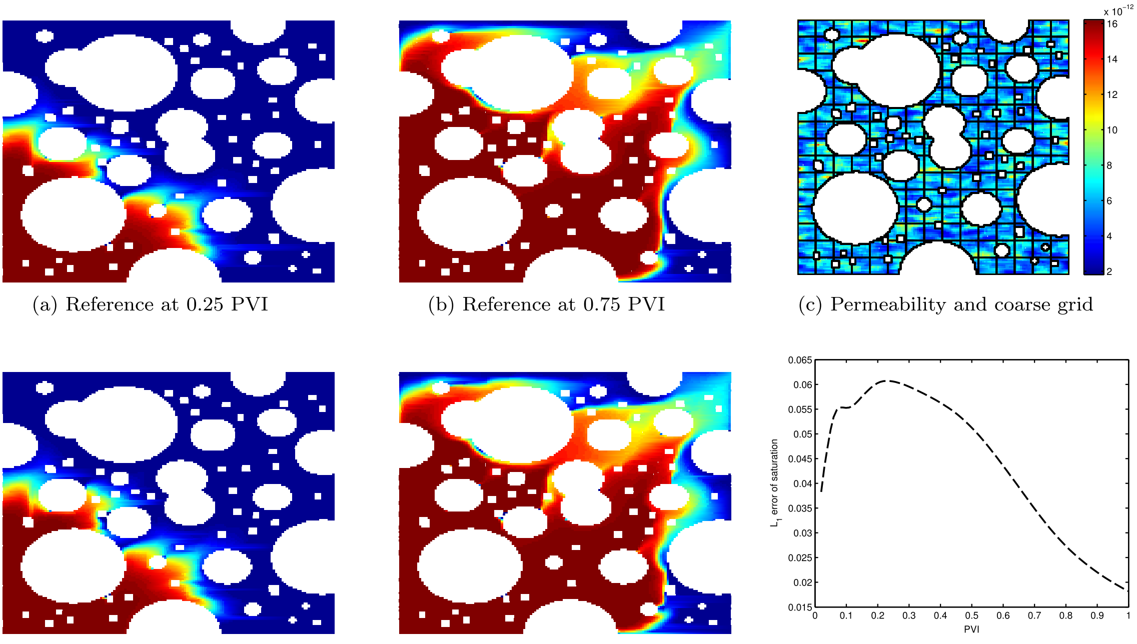
Fig. 18. The water saturation at different timesteps for the unstructured testcase with local refinement. The error colormap has a maximum of 0.5, which is higher than the largest error seen.

a total of two volumes. Adapting the restriction operator has no effect on the solution quality up to water breakthrough and only a minor positive effect afterwards. If we instead adapt the prolongation operator by fitting the coarse grid to the low-permeable barriers, the accuracy is significantly improved at the expense of having to work with a much more complicated coarse partition. In principle, one could combine the two approaches and adapt the prolongation operator *a priori* and the restriction operator *a posteriori*. Our experience is that this is difficult in practice as the combination of the two methods tends to produce control volumes that are highly convoluted and hence may cause singular or near-singular coarse problems.

Although we have seen several barrier cases for which adapting the restriction operator gives better results than adapting the prolongation operator, it is our general experience that barriers are best treated by adapting the coarse partition that defines the prolongation operator. However, one should be careful to not create overly complex and irregular grids that may introduce numerical artifacts and instabilities that may impact the approximation properties of the otherwise relatively robust MsRSB prolongation operator.

4.3.4. Unstructured grid adapted to faults and wells

The Gullfaks example showed how the structured corner-point format easily can result in unstructured topologies near faults and eroded or inactive cells. However, fully unstructured grids can also be of interest in regions without complex features, for instance to avoid consistency issues with the standard two-point discretization. In this example, we consider a two-dimensional PEBI mesh as shown in Fig. 17. The grid is adapted to local features, here exemplified by two sealing



(a) Reference at 0.25 PVI (b) Reference at 0.75 PVI (c) Permeability and coarse grid

(d) MsRSB at 0.25 PVI (e) MsRSB at 0.75 PVI (f) Saturation error in L_1 norm

Fig. 19. Saturation fronts for a model containing a large number of impermeable zones throughout the domain modeled as inactive cells.

faults and two wells. The cells near the faults are oriented to that the cell faces follow the fault and near the well we have introduced a radial grid refinement to better approximate the radial flow in the near-well zone.

The fluid model is a simple incompressible fluid with Corey-type relative permeability curves with quadratic exponents. The reservoir is initially filled with oil having a viscosity of 5 centipoise. Water is injected from a well near the upper-left corner whereas fluids are produced at a constant rate from a well near the bottom-right corner, with no-flow conditions at the boundary. Petrophysical data are sampled from Layer 35 of the Tarbert formation from SPE 10 using nearest neighbor interpolation. The resulting grid has 3265 cells. We have selected a slightly smaller grid than the base case to make it easier to see the local refinements. This means that the petrophysical parameters may have somewhat stronger contrasts (more abrupt jumps) than specified in the original SPE 10 data set. The faults are considered to be completely sealing, i.e., the transmissibility over the interfaces is zero.

For the multiscale solver, we create a simple partition using the centroids of the cells. The coarse blocks are then split across faults and the coarse blocks in the radial subsgrids are partitioned using distance from the wellbore. Once all the local features are accounted for, the coarse grid itself is also unstructured and altogether this represents a challenging test case. The coarse grid has 146 blocks. Away from unstructured features the median number of cells in each block is 42.

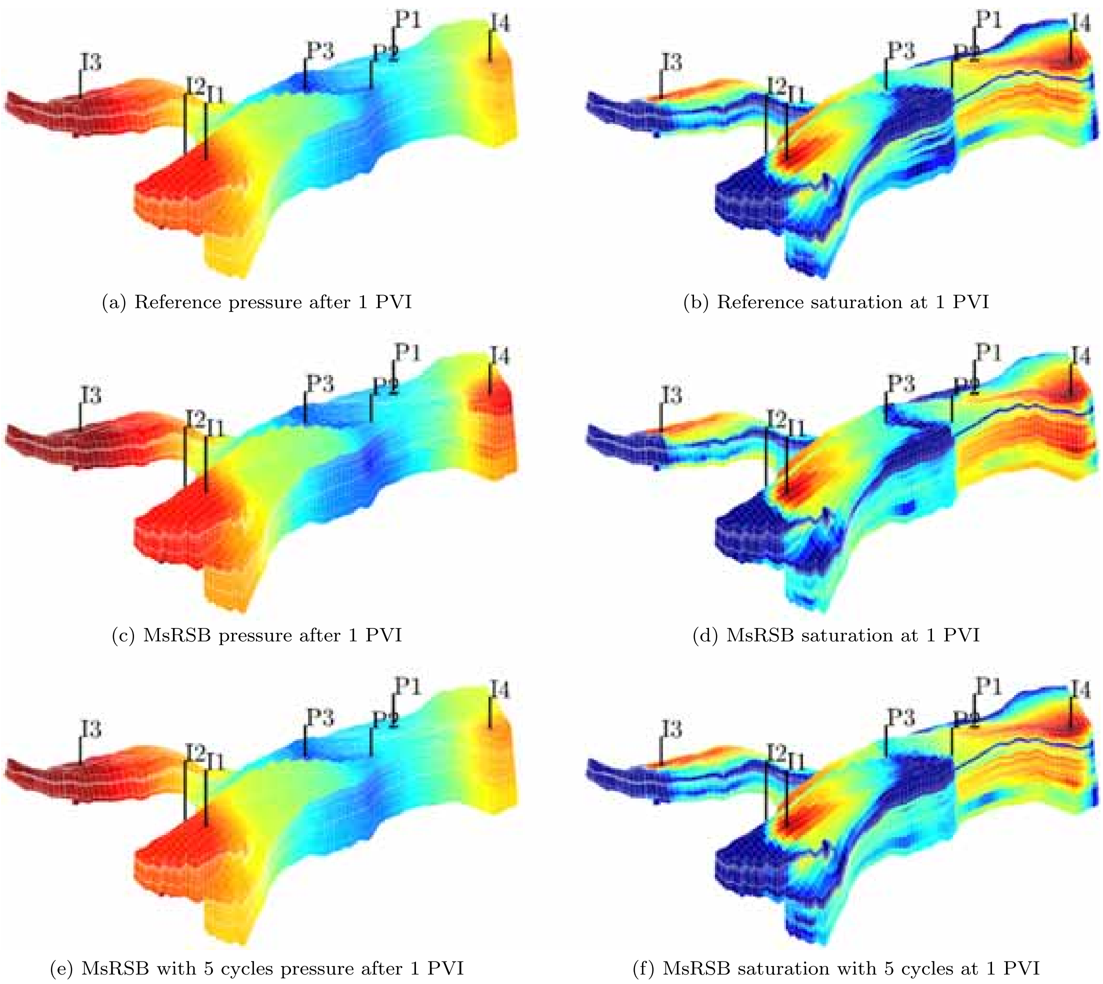
The saturation profiles for the reference and multiscale simulations presented in Fig. 18 show that the multiscale approximation captures the flow pattern accurately. Especially, the local fluid behavior near the faults is correctly represented; such local flow can easily be lost in a typical upscaled model.

4.3.5. Model with a large number of inactive cells

For many models there will be a large number of cells with insignificant permeability and/or porosity. There are different ways to treat such cells. One approach is to simply let the cells retain their small values and include them in the full system. By including the cells, however, the computational effort will be just as large as if they were highly permeable even though nothing will flow through them. The other option is to remove cells with impermeable rock types, which is what we will consider here. Consider a 1×1 km domain comprised of 150×150 fine cells, with lognormal permeability as shown in Fig. 19. We remove 43% of the cells, place a quarter five-spot well pattern with injection in the lower-left corner and proceed to inject one pore volume over ten years with unit mobility ratio. The multiscale solver uses a 15×15 uniform coarse grid and gives an accurate reproduction of the injected fluid, with a flux error of 0.15 in the L_2 norm.

4.3.6. Norne field model

In our final example, we study a water-injection problem posed on a geological model with grid geometry and petrophysical properties taken from the Norne benchmark case [60]. Initially, the reservoir is completely filled with oil, and the fluid behavior is described using an incompressible, two-phase model with the same fluid properties as in Section 4.3.4. The



(a) Reference pressure after 1 PVI

(b) Reference saturation at 1 PVI

(c) MsRSB pressure after 1 PVI

(d) MsRSB saturation at 1 PVI

(e) MsRSB with 5 cycles pressure after 1 PVI

(f) MsRSB saturation with 5 cycles at 1 PVI

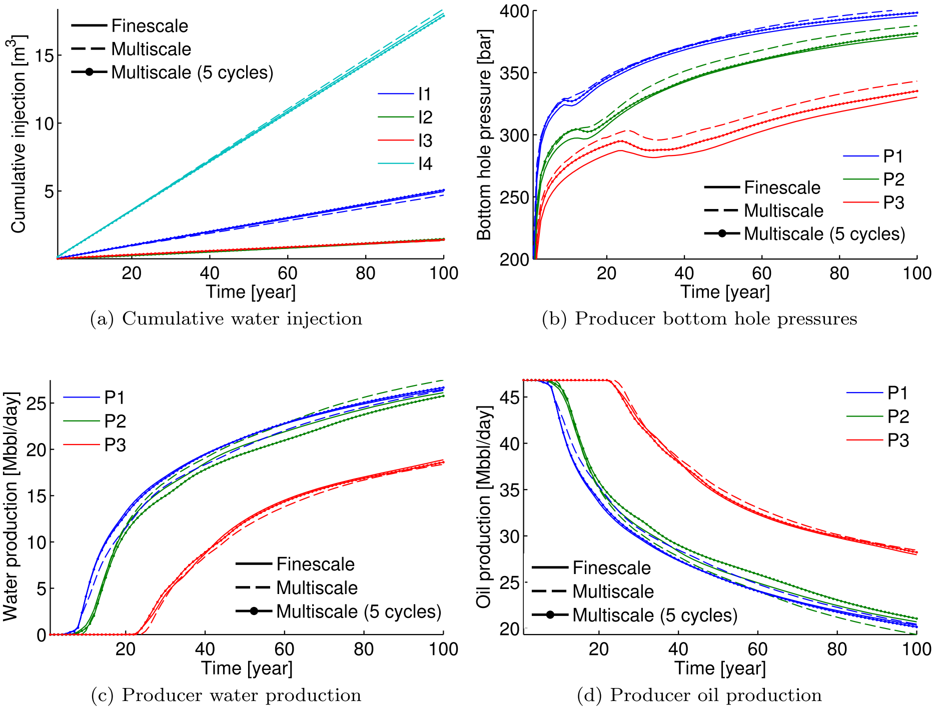
Fig. 20. Pressure and saturation profiles after the injection of one pore-volume of water for the model derived from the Norne benchmark data. All plots of the same type use the same color axes.

grid has 44 420 fine cells and layered permeability distribution. The model consists of two disconnected rock formations that are separated by full layer of inactive cells, so that the wells are the primary means of communication between the top three layers and the remaining parts of the model. Several faults and partially eroded layers makes the model effectively unstructured.

The reservoir has seven wells, three producers and four injectors. Each well is completed in all vertical layers, and will hence pass through multiple coarse blocks. The injectors operate at fixed pressure of 500 bar, while the producers have a constant rate corresponding to the drainage of a complete pore volume over the production period of 100 years.

The grid is partitioned into 250 coarse blocks using Metis with transmissibilities as edge weights. Wells are resolved most accurately in a multiscale method if they are placed near the centers of the coarse blocks. Making coarse grids that satisfy this property is straightforward if the grid has a simple structure and each well only perforate a single cell or a small contiguous set of cells. For complex grids with multiple perforations per well, however, it is generally not feasible to make such grids. We therefore consider two different set-ups for the multiscale solver: In the first, we use one pass of the multiscale solver to compute an approximate pressure. In the second, we use five multiscale cycles that each consist of a multiscale solve followed by five Jacobi iterations to remove local error and account for near-well information.

Fig. 20 shows pressure and saturation profiles after the injection of one pore-volume of water, as computed by the fine-scale solver and the two multiscale solver. We note that the multiscale pressure approximation is in close agreement with the fine-scale solution, irregardless of whether we use iterations or not. For the saturation field, the multiscale solvers exhibit some smearing near the boundary of the domain, but otherwise reproduce the water distribution at the end of simulation excellently. This is also confirmed by the well curves reported in Fig. 21. Even without iterations, the multiscale



(a) Cumulative water injection

(b) Producer bottom hole pressures

(c) Producer water production

(d) Producer oil production

Fig. 21. Well curves for the water-injection test case based on the Norne benchmark data.

solver is able to correctly reproduce the qualitative behavior of the producer, with a discrepancy of less than 5% in the bottom-hole pressure and fluid rates. By adding more iterations, this discrepancy can be reduced to the point where one also has an excellent quantitative match.

5. Concluding remarks

We have presented a novel multiscale formulation that relies on an iterative process to construct basis functions. The iterative process is defined as standard matrix manipulations in combination with simple grid indicators. These indicators can easily be obtained from general coarse partitions without any stringent requirements on the underlying grid topology. Because the formulation only requires knowledge of cell centroids and topological neighborship, the method is applicable to unstructured polyhedral grids in general and stratigraphic grids in particular. The method is flexible and very robust with respect to the shape of the coarse blocks, and it is therefore easy to formulate automated strategies to coarsen complex grid models having cells with degenerate geometry and unstructured topology caused by erosion, faults, inactive cells, local grid refinement, and other types of non-neighboring connections. In particular, the coarse partition can be adapted to geological features or to provide extra resolution, e.g., in near-well regions. This can be done in two ways, either by fitting the coarse grid *a priori* to adapt the prolongation operator or by modifying the control volumes *a posteriori* to adapt the restriction operator.

Extensive numerical tests on single-phase problems, some of which are reported herein and in [43], show that the MsRSB method compares favorably with the classical MsFV in terms of accuracy, and is generally much more robust and less affected by the choice of restriction operator and the resolution and type of the coarse partition. Notably, the pressure field for the challenging SPE 10 benchmark is solved to within 5% accuracy without the need for additional iterations or coarse grid adaption. Likewise, a high degree of accuracy is observed on models of the Norne field and the very structurally complex Gullfaks field. For both models, a wide range of coarse partitions can easily be generated using standard graph partitioning algorithms. This indicates that automated coarsening is indeed possible for realistic models with high media contrasts, complex unstructured topology, and cells with very high aspect ratios and degenerate geometries.

Through a set of multiphase test cases we have also demonstrated that the method can easily adapt to changing mobilities and provide prediction of well curves that are qualitatively correct, also when wells crossing different geological layers are completed in multiple coarse blocks. By setting tighter tolerances on the fine-scale residuals, excellent quantitative match can also be obtained at the cost of a few additional iterations. In many aspects, the iterative version of the method resembles an agglomeration-based multigrid method, but unlike these methods, MsRSB can be stopped at any prescribed residual tolerance and still produce a mass-conservative approximation.

Acknowledgements

This research was funded in part by Schlumberger Information Solutions and by the Research Council of Norway under grant no. 226035. We thank Jostein R. Natvig and Halvor Møll Nilsen for constructive discussions. We also thank Statoil (operator of the Norne field) and its license partners ENI and Petoro for the release of the Norne data and acknowledge the Center for Integrated Operations at NTNU for cooperation and coordination of the Norne cases.

References

[1] Y. Efendiev, T.Y. Hou, Multiscale Finite Element Methods, Surveys and Tutorials in the Applied Mathematical Sciences, vol. 4, Springer Verlag, New York, 2009.

[2] I. Babuška, G. Caloz, J. Osborn, Special finite element methods for a class of second order elliptic problems with rough coefficients, SIAM J. Numer. Anal. 31 (4) (1994) 945–981, http://dx.doi.org/10.1137/0731051.

[3] T. 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, http://dx.doi.org/10.1006/jcph.1997.5682.

[4] T. Arbogast, Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow, Comput. Geosci. 6 (3–4) (2002) 453–481, http://dx.doi.org/10.1023/A:1021295215383.

[5] T. Arbogast, K.J. Boyd, Subgrid upscaling and mixed multiscale finite elements, SIAM J. Numer. Anal. 44 (3) (2004) 1150–1171.

[6] Z. Chen, T.Y. Hou, A mixed multiscale finite element method for elliptic problems with oscillating coefficients, Math. Comput. 72 (2003) 541–576, http://dx.doi.org/10.1090/S0025-5718-02-01441-2.

[7] V. Kippe, J.E. Aarnes, K.-A. Lie, Multiscale finite-element methods for elliptic problems in porous media flow, in: P. Binning, P. Engesgaard, H. Dahle, G. Pinder, W. Gray (Eds.), Proceedings of the XVI International Conference on Computational Methods in Water Resources, Copenhagen, Denmark, 2006, http://proceedings.cmwr-xvi.org/.

[8] 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, http://dx.doi.org/10.1016/S0021-9991(03)00075-5.

[9] T. Arbogast, G. Pencheva, M.F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, Multiscale Model. Simul. 6 (1) (2007) 319–346, http://dx.doi.org/10.1137/060662587.

[10] K. Lipnikov, J.D. Moulton, D. Svyatskiy, A multilevel multiscale mimetic (m^3) method for two-phase flows in porous media, J. Comput. Phys. 227 (14) (2008) 6727–6753, http://dx.doi.org/10.1016/j.jcp.2008.03.029.

[11] J.E. Aarnes, On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation, Multiscale Model. Simul. 2 (3) (2004) 421–439, http://dx.doi.org/10.1137/030600655.

[12] J. Aarnes, K.-A. Lie, Toward reservoir simulation on geological grid models, in: Proceedings of the 9th European Conference on the Mathematics of Oil Recovery, EAGE, Cannes, France, 2004.

[13] I. Lunati, P. Jenny, Multiscale finite-volume method for density-driven flow in porous media, Comput. Geosci. 12 (3) (2008) 337–350, http://dx.doi.org/10.1007/s10596-007-9071-9.

[14] I. Lunati, P. Jenny, Multiscale finite-volume method for compressible multiphase flow in porous media, J. Comput. Phys. 216 (2) (2006) 616–636, http://dx.doi.org/10.1016/j.jcp.2006.01.001.

[15] S.H. Lee, C. Wolfsteiner, H. Tchelepi, Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three phase flow with gravity, Comput. Geosci. 12 (3) (2008) 351–366, http://dx.doi.org/10.1007/s10596-007-9069-3.

[16] H. Zhou, H.A. Tchelepi, Operator-based multiscale method for compressible flow, SPE J. 13 (2) (2008) 267–273, http://dx.doi.org/10.2118/106254-PA.

[17] H. Hajibeygi, P. Jenny, Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media, J. Comput. Phys. 228 (14) (2009) 5129–5147, http://dx.doi.org/10.1016/j.jcp.2009.04.017.

[18] H. Hajibeygi, H.A. Tchelepi, Compositional multiscale finite-volume formulation, SPE J. 19 (2) (2014) 316–326, http://dx.doi.org/10.2118/163664-PA.

[19] I. Lunati, M. Tyagi, S.H. Lee, An iterative multiscale finite volume algorithm converging to the exact solution, J. Comput. Phys. 230 (5) (2011) 1849–1864, http://dx.doi.org/10.1016/j.jcp.2010.11.036.

[20] H. Zhou, H.A. Tchelepi, Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, SPE J. 17 (2) (2012) 523–539, http://dx.doi.org/10.2118/141473-PA.

[21] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Algebraic multiscale solver for flow in heterogeneous porous media, J. Comput. Phys. 259 (2014) 284–303, http://dx.doi.org/10.1016/j.jcp.2013.11.024.

[22] O. Møyner, Multiscale finite-volume methods on unstructured grids, Master's thesis, Norwegian University of Science and Technology, Trondheim, 2012, http://daim.idi.ntnu.no/masteroppgave?id=7377.

[23] O. Møyner, K.-A. Lie, The multiscale finite volume method on unstructured grids, in: SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 18–20 February 2013, 2013, sPE 163649-MS.

[24] T.H. Sandve, I. Berre, E. Keilegavlen, J.M. Nordbotten, Multiscale simulation of flow and heat transport in fractured geothermal reservoirs: inexact solvers and improved transport upscaling, in: Thirty-Eighth Workshop on Geothermal Reservoir Engineering, Stanford University, February 11–13, Stanford, California, USA, 2013.

[25] O. Møyner, K.-A. Lie, The multiscale finite-volume method on stratigraphic grids, SPE J. 19 (5) (2014) 816–831, http://dx.doi.org/10.2118/163649-PA.

[26] A. Kozlova, Z. Li, J.R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, S.H. Lee, A real-field multiscale black-oil reservoir simulator, in: SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas, USA, 2015, sPE 173226-MS.

[27] E. Parramore, M.G. Edwards, S. Lamine, M. Pal, Multiscale formulations with CVD-MPFA schemes on structured and unstructured grids, in: SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 18–20 February 2013, 2013, sPE 163626-MS.

[28] J.E. Aarnes, S. Krogstad, K.-A. Lie, Multiscale mixed/mimetic methods on corner-point grids, Comput. Geosci. 12 (3) (2008) 297–315, http://dx.doi.org/10.1007/s10596-007-9072-8.

[29] J.R. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, S.K. Khataniar, Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs, SPE J. 16 (4) (2011) 880–888, http://dx.doi.org/10.2018/119132-PA.

[30] F.O. Alpak, M. Pal, K.-A. Lie, A multiscale method for modeling flow in stratigraphically complex reservoirs, SPE J. 17 (4) (2012) 1056–1070, http://dx.doi.org/10.2118/140403-PA.

[31] M. Pal, S. Lamine, K.-A. Lie, S. Krogstad, Multiscale method for simulating two and three-phase flow in porous media, in: SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18–20 February 2013, 2013, sPE 163669-MS.

[32] S. Krogstad, K.-A. Lie, H.M. Nilsen, J.R. Natvig, B. Skaflestad, J.E. Aarnes, A multiscale mixed finite-element solver for three-phase black-oil flow, in: SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009, 2009.

[33] S. Krogstad, K.-A. Lie, B. Skaflestad, Mixed multiscale methods for compressible flow, in: Proceedings of ECMOR XIII—13th European Conference on the Mathematics of Oil Recovery, EAGE, Biarritz, France, 2012.

[34] O. Møyner, K.-A. Lie, A multiscale two-point flux-approximation method, J. Comput. Phys. 275 (2014) 273–293, http://dx.doi.org/10.1016/j.jcp.2014.07.003.

[35] P. Vanek, J. Mandel, M. Brezina, Algebraic Multigrid on Unstructured Meshes, Tech. rep. 34, University of Colorado at Denver, Denver, CO, USA, 1994.

[36] P. Vanek, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing 56 (3) (1996) 179–196, http://dx.doi.org/10.1007/BF02238511.

[37] P. Vanek, M. Brezina, J. Mandel, Convergence of algebraic multigrid based on smoothed aggregation, Numer. Math. 88 (3) (2001) 559–579, http://dx.doi.org/10.1007/s211-001-8015-y.

[38] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, J. Ruge, Adaptive smoothed aggregation (α sa) multigrid, SIAM Rev. 47 (2) (2005) 317–346.

[39] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (1) (2001) 281–309.

[40] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, PETSc users manual, Tech. rep. ANL-95/11 – Revision 3.6, Argonne National Laboratory, 2015, http://www.mcs.anl.gov/petsc.

[41] M. Gee, C. Siefert, J. Hu, R. Tuminaro, M. Sala, ML 5.0 smoothed aggregation user's guide, Tech. rep. SAND2006-2649, Sandia National Laboratories, 2006.

[42] O. Møyner, K.-A. Lie, A multiscale restriction-smoothed basis method for compressible black-oil models, SPE J., submitted for publication.

[43] O. Møyner, Construction of multiscale preconditioners on stratigraphic grids, in: ECMOR XIV – 14th European Conference on the Mathematics of Oil Recovery, Catania, Sicily, Italy, 8–11 September 2014, EAGE, 2014.

[44] Y. Brenier, J. Jaffré, Upstream differencing for multiphase flow in reservoir simulation, SIAM J. Numer. Anal. 28 (3) (1991) 685–696, http://dx.doi.org/10.1137/0728036.

[45] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, SPE Reserv. Eval. Eng. 4 (2001) 308–317, http://dx.doi.org/10.2118/72469-PA, http://www.spe.org/csp/.

[46] I. Lunati, S.H. Lee, An operator formulation of the multiscale finite-volume method with correction function, Multiscale Model. Simul. 8 (1) (2009) 96–109, http://dx.doi.org/10.1137/080742117.

[47] K.-A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H. Nilsen, B. Skaflestad, Open-source MATLAB implementation of consistent discretisations on complex grids, Comput. Geosci. 16 (2012) 297–322, http://dx.doi.org/10.1007/s10596-011-9244-4.

[48] S. Krogstad, K.-A. Lie, O. Møyner, H.M. Nilsen, X. Raynaud, B. Skaflestad, MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems, in: SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas, 2015.

[49] K.-A. Lie, An introduction to reservoir simulation using MATLAB: user guide for the Matlab reservoir Simulation Toolbox (MRST), SINTEF ICT, http://www.sintef.no/Projectweb/MRST/publications, May 2014, 1st edition.

[50] The MATLAB reservoir simulation toolbox, version 2014b, http://www.sintef.no/MRST/, Nov. 2014.

[51] A. Sandvin, E. Keilegavlen, J.M. Nordbotten, Auxiliary variables for 3d multiscale simulations in heterogeneous porous media, J. Comput. Phys. 238 (2013) 141–153, http://dx.doi.org/10.1016/j.jcp.2012.12.016.

[52] H. Fossen, J. Hesthammer, Structural geology of the Gullfaks Field, in: M.P. Coward, H. Johnson, T.S. Daltaban (Eds.), Structural Geology in Reservoir Characterization, vol. 127, Geological Society Special Publication, 1998, pp. 231–261.

[53] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1998) 359–392, http://dx.doi.org/10.1137/S1064827595287997.

[54] V. Kippe, J.E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Comput. Geosci. 12 (3) (2008) 377–398, http://dx.doi.org/10.1007/s10596-007-9074-6.

[55] M.A. Hesse, B.T. Mallison, H.A. Tchelepi, Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations, Multiscale Model. Simul. 7 (2) (2008) 934–962, http://dx.doi.org/10.1137/070705015.

[56] I. Lunati, P. Jenny, Treating highly anisotropic subsurface flow with the multiscale finite-volume method, Multiscale Model. Simul. 6 (1) (2007) 308–318, http://dx.doi.org/10.1137/050638928.

[57] M. Tene, M.S. Al Kobaisi, H. Hajibeygi, Algebraic multiscale solver for flow in heterogeneous fractured porous media, in: SPE Reservoir Simulation Symposium Held in Houston, Texas, USA, 23–25 February 2015, 2015, sPE 173200-MS.

[58] J.E. Aarnes, S. Krogstad, K.-A. Lie, A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids, Multiscale Model. Simul. 5 (2) (2006) 337–363, http://dx.doi.org/10.1137/050634566.

[59] K.-A. Lie, J.R. Natvig, S. Krogstad, Y. Yang, X.-H. Wu, Grid adaptation for the Dirichlet–Neumann representation method and the multiscale mixed finite-element method, Comput. Geosci. 18 (3) (2014) 357–372, http://dx.doi.org/10.1007/s10596-013-9397-4.

[60] IO Center – Norne Benchmark Case, http://www.ipt.ntnu.no/~norne/wiki/doku.php, Jun. 2012.

Paper VI

Application of Flow Diagnostics and Multiscale Methods for Reservoir Management

Knut-Andreas Lie, Olav Møyner and Stein Krogstad

In proceedings of the 2015 SPE Reservoir Simulation Symposium,
Houston, Texas, USA
DOI: 10.2118/173306-MS

Paper VII

MRST-AD – An Open-Source Framework for Rapid Prototyping and Evaluation of Reservoir Simulation Problems

Stein Krogstad, Knut–Andreas Lie, Olav Møyner, Halvor Møll Nilsen,
Xavier Raynaud and Bård Skaflestad

*In proceedings of the 2015 SPE Reservoir Simulation Symposium,
Houston, Texas, USA
DOI: 10.2118/173317-MS*

Paper VIII

A Multiscale Restriction-Smoothed Basis Method for Compressible Black-Oil Models

Olav Møyner and Knut-Andreas Lie

SPE Journal, published ahead of print, June 2016
DOI: 10.2118/173265-PA

Is not included due to copyright

Paper IX

Multiscale Simulation of Polymer Flooding with Shear Effects

Sindre Hilden, Olav Møyner, Knut-Andreas Lie and Kai Bao

Transport in Porous Media, volume 113, issue 1, pp. 111–135, 2016
DOI: 10.1007/s11242-016-0682-2

Paper X

The Multiscale Restriction Smoothed Basis Method for Fractured Porous Media (F-MsRSB)

Swej Shah, Olav Møyner, Matei Tene, Knut-Andreas Lie and Hadi Hajibeygi

Journal of Computational Physics, volume 318, pp. 36–57, 2016
DOI: 10.1016/j.jcp.2016.05.001

Journal of Computational Physics 318 (2016) 36–57

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

The multiscale restriction smoothed basis method for fractured porous media (F-MsRSB)

Swej Shah [a,*], Olav Møyner [b], Matei Tene [a], Knut-Andreas Lie [b], Hadi Hajibeygi [a]

[a] *Department of Geoscience and Engineering, Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands*

[b] *SINTEF, Department of Applied Mathematics, Forskningsveien 1, P.O. Box 124 Blindern, N-0314 Oslo, Norway*

ARTICLE INFO

Article history:
Received 1 October 2015
Received in revised form 29 March 2016
Accepted 1 May 2016
Available online 4 May 2016

Keywords:
Fractured porous media
Embedded fractured modeling
Multiscale finite-volume methods
Restriction smoothed basis
Algebraic iterative multiscale solver
Scalable linear solvers
Multiphase flow
Reservoir simulation

ABSTRACT

A novel multiscale method for multiphase flow in heterogeneous fractured porous media is devised. The discrete fine-scale system is described using an embedded fracture modeling approach, in which the heterogeneous rock (matrix) and highly-conductive fractures are represented on independent grids. Given this fine-scale discrete system, the method first partitions the fine-scale volumetric grid representing the matrix and the lower-dimensional grids representing fractures into independent coarse grids. Then, basis functions for matrix and fractures are constructed by restricted smoothing, which gives a flexible and robust treatment of complex geometrical features and heterogeneous coefficients. From the basis functions one constructs a prolongation operator that maps between the coarse- and fine-scale systems. The resulting method allows for general coupling of matrix and fracture basis functions, giving efficient treatment of a large variety of fracture conductivities. In addition, basis functions can be adaptively updated using efficient global smoothing strategies to account for multiphase flow effects. The method is conservative and because it is described and implemented in algebraic form, it is straightforward to employ it to both rectilinear and unstructured grids. Through a series of challenging test cases for single and multiphase flow, in which synthetic and realistic fracture maps are combined with heterogeneous petrophysical matrix properties, we validate the method and conclude that it is an efficient and accurate approach for simulating flow in complex, large-scale, fractured media.

1. Introduction

Accurate simulation of multiphase flow in natural porous media represented on high-resolution numerical grids is computationally demanding. Fine-scale petrophysical properties like permeability are often highly heterogeneous, change over several orders of magnitudes, and, in general, do not entail scale separation [1]. This computational challenge has motivated the development of several multiscale methods, which solve accurate coarse-scale systems constructed by the use of locally-computed basis functions [2–13]. Once the coarse-scale system is solved, its solution is interpolated into the original fine-scale resolution using the sub-resolution of the basis functions. Among the proposed multiscale methods, multiscale finite-volume (MSFV) methods not only provide mass-conservative solutions at fine-scale, which is a crucial property for

[*] Corresponding author.
E-mail addresses: S.Y.Shah@tudelft.nl (S. Shah), Olav.Moyner@sintef.no (O. Møyner), M.Tene@tudelft.nl (M. Tene), Knut-Andreas.Lie@sintef.no (K.-A. Lie), H.Hajibeygi@tudelft.nl (H. Hajibeygi).

http://dx.doi.org/10.1016/j.jcp.2016.05.001
0021-9991/

convergent solution of transport equations, but also enable relatively simple inclusion of the type of multiphase flow equations seen in contemporary reservoir models [3,14–19].

Multiscale methods compute approximate solutions having the original fine-scale resolution so that their error (or residual) can be calculated with respect to the fine-scale discrete system. As such, one can achieve systematic strategies for reducing the error through iterative procedures that combine the multiscale solver with a fine-scale smoother [20–23]. Iterative multiscale methods are scalable and deliver mass-conservative solutions after any MSFV stage. The latter property makes them unique compared with alternative advanced solvers, such as multigrid methods [24]. Recent developments of the MSFV method include extensions to compressible and compositional nonlinear displacements [25,26], unstructured grids [11,27] and fully-implicit simulations [28]. While these important developments, combined, cast a promising framework for next-generation simulators, they have been focused mainly on addressing challenges due to complex fluid physics, highly heterogeneous rock properties, and complex computational mesh geometries.

Many geological formations—including hydrocarbon reservoirs, underground water resources, and geothermal energy production fields—are naturally fractured. Fractures are highly conductive channels which, for most practical purposes, exist in a lower-dimensional space compared to the porous matrix. Physical properties inside fractures and their length scales can be very different from those of the surrounding rock, adding significantly to the computational challenges, especially once realistic length scales and complex fracture network maps are considered. As a result, a variety of modeling approaches and numerical methods for different types of fractured reservoirs have been proposed [29–42,40,43,44]. Among them, the embedded fracture modeling approach [35,36,45,46,34,47] benefits from independent grids for fracture and matrix, and constitutes a promising approach for naturally fractured reservoirs and also for cases with dynamic fracture creations and closure of, e.g., geothermal systems. Note that small-scale fractures (smaller than fine-scale grid cells) are homogenized within the matrix porous rock, forming effective matrix conductivities [34]. This approach, similar to other discrete-fracture-modeling (DFM) approaches, leads to detailed fine-scale discrete systems (for matrix and fracture unknowns) with high contrasts within the entries, which are clearly much more challenging to be solved efficiently than non-fractured heterogeneous cases. Therefore, it is highly important to develop efficient multiscale methods for fractured formations.

Early attempts at developing multiscale methods for fractured media were based on a mixed finite-element formulation in which high-conductive fractures were either represented explicitly as volumetric objects [37] or the fracture–matrix interaction was modeled by the Stokes–Brinkmann equations [38,48]. Within the MSFV framework, Hajibeygi et al. [47] developed the first multiscale method for fractured porous media, in which additional fracture basis functions were introduced to map each fracture network into one coarse-scale degree-of-freedom (DOF). Later, Sandve et al. [49] used the MSFV method to develop effective coarse-scale MINC-type model for fracture networks. Very recently, Tene et al. [50,51] developed a general formulation for fractured media by proposing an algebraic multiscale solver for fractured media (F-AMS). In the F-AMS, fracture basis functions were introduced on the basis of a coarsening ratio inside fracture domain, similar as in the matrix rock. Results of F-AMS when only a few fracture DOFs were used illustrated that such a multiscale map for fractured domains is quite efficient. Similar to all MSFV and AMS methods, F-AMS relies on coarse and dual-coarse grids imposed on the provided fine-scale grid cells. While the former is used to construct mass-conservative, coarse-scale systems, the latter is employed to compute local basis functions. However, geological complexities and the use of complex grid geometries make the construction of these two coarse grids quite challenging. Recently, the multiscale restriction smoothed basis (MsRSB) method was devised to overcome this complexity [52]. The MsRSB is unique in the way the basis functions are computed, yet leads to a stable and robust treatment of complex heterogeneous coefficients [53], as well as realistic flow physics for improved and enhanced oil recovery [54,55]. It is therefore favorable to use this method as a basis when seeking to extend multiscale simulation approaches for more complex fractured media.

In this work, a multiscale restricted smoothed basis method for fractured media (F-MsRSB) is developed. Following F-AMS [50,51], F-MsRSB constructs basis functions for fractures and matrix in a general way, allowing for different level of coupling between them. In addition to F-AMS, though, F-MsRSB constructs its multiscale formulation on the basis of the MsRSB approach. This would facilitate its extensions toward complex geometries while maintaining its efficiency for highly heterogeneous challenging scenarios such as the SPE10 comparative test case [56]. Unlike previous works, the performance of F-MsRSB is investigated for realistic fracture models with complex fracture networks. Transmissibility-weighted connectivity graphs of independent fractures are decomposed using the METIS software [57], leading to an automatic coarsening strategy for fractures. Following the traditional algebraic multiscale formulations, F-MsRSB can easily be adapted to account for complex physics such as compressibility [23] and gravity [58] as discussed in [54] for the MsRSB method. To facilitate implementation, specially for complex fracture networks, here, fracture cells are introduced into the discrete systems through non-neighboring connections (NNC).

Through several two- and three-dimensional cases with highly heterogeneous coefficients, F-MsRSB is found to efficiently compute approximate solutions of good quality. Furthermore, in order to allow for error control and reduction strategies, especially for multiphase flow scenarios, the method is combined with a fine-scale smoother, ILU(0) [59,22,23]. While low-frequency errors are resolved by the coarse-scale system in F-MsRSB, the fine-scale smoother resolves high-frequency errors, the combination of which leads to an efficient iterative multiscale solver for fractured media. These iterations are applied adaptively and infrequently just to maintain user-prescribed accuracy. Several multiphase flow cases are considered in which the adaptive iterative F-MsRSB is employed to efficiently compute high-quality solutions for the flow equations. All of these systematic single- and multiphase flow cases reveal that F-MsRSB is an efficient and versatile multiscale method for naturally fractured reservoirs with highly heterogeneous coefficients.

The paper is structured as follows. The fine-scale discrete system for flow in fractured porous media is described in Section 2. Then, in Section 3.1 and Section 3.3 AMS and MsRSB are revisited, respectively. The development of F-MsRSB is presented in Section 4. Numerical results for single- and multiphase flow for both 2D and 3D heterogeneous reservoirs are presented in Section 5. Finally, the paper is concluded in Section 6.

2. Governing equations and fine-scale system

Mass conservation for n_{ph} incompressible phases flowing in a porous medium reads

$$\frac{\partial}{\partial t}(\phi S_\alpha) - \nabla \cdot (\boldsymbol{\lambda}_\alpha \cdot \nabla p) = q_\alpha \quad \forall \alpha \in \{1, \dots, n_{ph}\}, \tag{1}$$

where Darcy's law is employed to replace phase velocity u_α with pressure gradient ∇p. Here, gravitational and capillary effects are both neglected. Moreover, S_α and $\boldsymbol{\lambda}_\alpha$ are phase saturation and mobility, respectively. Note that $\boldsymbol{\lambda}_\alpha = \mathbf{k} k_{r\alpha}/\mu_\alpha$ holds, where the positive-definite permeability tensor $\mathbf{k}$ is typically highly heterogeneous at multiple scales. Also, relative permeability, $k_{r\alpha}$ and phase viscosity, μ_α, are given functions of primary unknowns p and S. These balance equations, along with the constraint that all phases fill the pore volume, i.e.,

$$\sum_{\alpha=1}^{n_{ph}} S_\alpha = 1, \tag{2}$$

form a well-posed system of equations for $(n_{ph}+1)$ unknowns. Sequential approaches derive a pressure equation, which is solved first, then phase velocities are obtained to subsequently solve $n_{ph}-1$ transport equations (1). The n_{ph}-th saturation is obtained using the constraint (2). To obtain the pressure equation, i.e.,

$$-\nabla \cdot (\boldsymbol{\lambda}_t \cdot \nabla p) = q_t, \tag{3}$$

the mass-balance equations (1) are summed up and the time-dependent term (accumulation) cancels out owing to the constraint (2). Total mobility, $\boldsymbol{\lambda}_t$ and total source terms q_t are obtained by summing their phase-wise counterparts.

For fractured porous media, following the hierarchical fracture model approach, small-scale fractures are homogenized and represented by an effective matrix permeability $\mathbf{k}^m \in \mathbb{R}^n$, whereas fractures with larger length scales are explicitly represented with an embedded fracture modeling approach (EFM). Important is that the fracture elements can cross over matrix cells, or be confined at their interfaces. In the latter case, EFM reduces to alternative discrete fracture modeling approaches. Note that fractures are lower-dimensional manifolds owing to their small apertures, so that $\mathbf{k}^f \in \mathbb{R}^{n-1}$. In this case, the pressure equation can be expressed as

$$-\nabla \cdot (\boldsymbol{\lambda}_\mathbf{t} \cdot \nabla p)^m + \psi^{mf} = q_t^m \quad \text{on} \quad \Omega^m \subset \mathbb{R}^n, \tag{4}$$

$$-\nabla \cdot (\boldsymbol{\lambda}_\mathbf{t} \cdot \nabla p)^f + \psi^{fm} = q_t^f \quad \text{on} \quad \Omega^f \subset \mathbb{R}^{n-1}, \tag{5}$$

where superscripts m and f represent matrix and fracture quantities, respectively. Mass exchange between fracture and matrix cells, ψ^{mf} and ψ^{fm}, is modeled as

$$\int_A \psi^{fm} dA = C_I \, \boldsymbol{\lambda}_t^* (p^f - p^m) = -\int_V \psi^{mf} dV, \tag{6}$$

where C_I is the fracture–matrix conductivity index [47,36]. Interaction of a matrix element i and a fracture element j is defined as

$$C_I = A_{i-j}/\langle d \rangle_{i-j}, \tag{7}$$

where A_{i-j} is the fracture plate area and $\langle d \rangle_{i-j}$ is the average normal distance between i and j. More information about embedded fracture models and the calculation of their parameters can be found in [47]. An advantage of EFM is that the fracture and matrix grids are independent and, thus, suited for many realistic scenarios such as naturally fractured reservoirs and dynamic fracture generation and closures.

Finite-volume discretization of (4) and (5) leads to a fine-scale system of equations, $Ap = q$, for matrix and fracture pressure unknowns, i.e.,

$$\begin{bmatrix} A_{mm} & A_{mf} & A_{mw} \\ A_{fm} & A_{ff} & A_{fw} \\ A_{wm} & A_{wf} & A_{ww} \end{bmatrix} \begin{bmatrix} p^m \\ p^f \\ p^w \end{bmatrix} = \begin{bmatrix} q^m \\ q^f \\ q^w \end{bmatrix}, \tag{8}$$

where super-index w denotes external well (source) terms [1]. Obviously, A_{fw} and A_{wf} will be zero if no well is drilled into the fracture domain.

The formulation of this paper is developed into the open-source MATLAB simulator MRST [60,61], in which fractures are introduced using non-neighboring connections (NNC) [42,62,41] and a sequentially-implicit strategy is used to simulate multiphase flow [53,54].

3. Multiscale restriction smoothed basis method (MsRSB)

Large-scale heterogeneous formations with complex fracture network maps, along with high contrasts between fracture and matrix properties make (8) quite challenging to solve using any classical numerical method. To resolve this computational challenge, a multiscale restriction smoothed basis method for fractured media (F-MsRSB) is developed. The F-MsRSB benefits from the previously developed multiscale methods for fractured media [47,50] and the MsRSB formulation for unfractured systems [52–54]. To describe the method, we start by discussing a general algebraic multiscale formulation [22,23] before we continue to describe the specific prolongation and restriction operators of the MsRSB method. This section will cast the foundation of the next section in which the novel development of this paper, i.e., F-MsRSB, is presented.

3.1. Algebraic multiscale formulation

To avoid solving (8) directly on the fine scale, multiscale methods introduce a prolongation operator $\mathcal{P}$ that maps between the degrees-of-freedom on the underlying fine-scale grid, that describes the (fractured) porous medium and its petrophysical parameters, and degrees-of-freedom associated with a coarse grid partition on which we will solve the global flow problem. In other words, if p^c and p' denote approximations on the coarse and fine grids, we have

$$p' = \mathcal{P} p^c. \tag{9}$$

Note that p' contains multiscale pressure approximations for both fracture and matrix at fine-grid resolution, i.e., $p' = [p'_m \ \ p'_f]^T$, while p^c contains coarse pressures for both matrix and fracture, i.e., $p^c = [p^{c,m} \ \ p^{c,f}]^T$ [50,51]. These two vectors contain $n_f = n_f^m + n_f^f$ and $n_c = n_{cm} + n_{cf}$ entries, where n_f and n_c are the numbers of fine and coarse cells, respectively, including both matrix and fractures. To determine p^c, we introduce a restriction operator $\mathcal{R}$ that maps the fine-scale system (8) into a coarse-scale system

$$\underbrace{(\mathcal{R}A\mathcal{P})}_{A^c} p^c = \underbrace{\mathcal{R}q}_{q^c}, \tag{10}$$

that has much smaller size (i.e., $n_c \times n_c$) than the original fine-scale system (i.e., $n_f \times n_f$), see [15]. Combining (10) and (9), the algebraic multiscale (AMS) procedure can be summarized as

$$p \approx p' = \underbrace{\mathcal{P}(\mathcal{R}A\mathcal{P})^{-1}\mathcal{R}}_{M_{ms}^{-1}} q. \tag{11}$$

In the same way, MsRSB imposes a coarse grid on top of the provided fine-scale grid. Inside each coarse grid block (coarse control volumes), a fine-scale grid cell is also selected as coarse node. Coarse blocks $\Omega_K^c \ \forall \ K \in \{1, \ldots, n_c\}$ define a non-overlapping partition of the domain, where n_c is the total number of coarse blocks in the system including n_{cm} matrix and n_{cf} fracture blocks.

For the restriction operator, there are two different choices, either to use $\mathcal{R} = \mathcal{P}^T$, which will lead to a Galerkin-type formulation, or to use a finite-volume restriction procedure [3], which can be stated as discrete integration operator over coarse control volumes Ω_K^c, i.e.,

$$\mathcal{R}_{i,K} = \begin{cases} 1, & \text{if } \mathbf{x}_i \subset \Omega_K^c, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Here $\mathbf{x}_i$ represents the i-th control volume at fine-scale. As shown in [52,53], the MsRSB method is not very sensitive to the choice of restriction operator and herein we use the finite-volume operator to ensure that we can reconstruct conservative fine-scale velocities.

The prolongation operator is constructed by solving localized flow problems, and the way these flow problems are set up varies from one method to another. However, in all multiscale methods, the prolongation operator $\mathcal{P}$ is defined so that it stores basis function Φ_K associated with coarse block Ω_K^c in its K-th column, i.e.,

$$\mathcal{P}_{i,K} = \Phi_K(\mathbf{x}_i) \quad \forall \, i \in \{1, \ldots, n_f\}, \, \forall \, K \in \{1, \ldots, n_c\}. \tag{13}$$

Here, $\Phi_K(\mathbf{x}_i)$ is the value of basis function Φ_K in the i-th fine-grid cell, $\mathbf{x}_i$. Both the original MSFV method [3] and its state-of-the-art extension (AMS) [22,23] rely on a secondary coarse partition, defined as the dual to the primal coarse grid, over which the basis functions Φ_K are locally computed. While it is possible to extend conservative multiscale methods based on a dual-grid formulation to stratigraphic and other types of unstructured grids [27,18,49,63], it has proved to be difficult, when possible, to develop satisfactory dual–primal partitions for grids with complex geometry. Moreover, localization errors induced by strong permeability contrasts across block boundaries introduce instabilities in the corresponding multipoint coarse-scale stencil. This motivated the development of a multiscale two-point flux-approximation formulation [11], in which an implicitly defined dual grid is used to compose elementary flow solutions into localized basis functions.

In the MsRSB method, however, local supports for basis functions are defined based on support regions, which are relatively simple to define even for very complex grids. Once these support regions are obtained, restriction-smoothed basis functions are computed by employing a modified form of the damped-Jacobi smoothing approach, similar as in smoothed-aggregation-based multigrid methods [64–66]. In the following sections, the MsRSB support regions and basis functions are briefly explained. Detailed explanations can be found in [52,53].

3.2. Coarse grid and support regions

Basis function Φ_K can have nonzero values only in the support region I_K. For the specific case of MSFV, e.g., I_K reduces to dual-coarse grid blocks. Thus, by construction, the basis function Φ_K and consequently the K-th column of $\mathcal{P}$ are set to zero outside I_K.

The support region of a coarse block Ω^c_K (see Fig. 1) is constructed by creating a local triangulation, using cell and shared-face centroids of all immediate geometrical neighbors of Ω^c_K [52]. This ensures that for a Cartesian grid without fractures, the coarse system has the same multipoint flux stencil as in the original MSFV method. Support regions for fractured coarse blocks are, because of their (potentially severe) geometrical complexities, computed by a different procedure, which will be described in the next section (see Algorithm 1).

The support boundary B_K is defined as the set of all fine cells that are topological neighbors of the outermost cells in the support region I_K. Note that $I_K \cap B_K = \emptyset$. This leads to the definition of a global boundary G which is a union of all B_K for all $K \in \{1, \dots, n_c\}$, i.e.,

$$G = B_1 \cup B_2 \cup \cdots \cup B_{n_c}. \tag{14}$$

Fig. 1 illustrates I, B and G for a 2D rectangular Cartesian and an unstructured hexagonal grid. For a Cartesian grid geometry, G becomes equivalent to the set of all dual-coarse boundary cells, i.e., similar to the classical MSFV method. Finally, indices of all support regions overlapping with each fine cell $\mathbf{x}_i$ stored inside G are stored in the set H_i, i.e.,

$$H_i = \{K | \mathbf{x}_i \in I_K, \mathbf{x}_i \in G\}. \tag{15}$$

H_i can be visualized using the last row of images in Fig. 1. Note that they follow the same indexing as depicted in Fig. 1a. A randomly picked fine cell belonging to the global boundary G, indexed as i, and denoted by the control volume $\mathbf{x}_i$ is depicted in blue in Fig. 1g. The next two images (Fig. 1h and Fig. 1i) show the support regions that encompass this control volume. Hence, the set H_i for this particular cell can be written as $H_i = \{6, 9\}$. This can be repeated for each fine cell stored inside G to generate the complete set H.

3.3. MsRSB prolongation operator

As mentioned earlier, basis functions are calculated iteratively, having nonzero values only inside their support regions. The basis functions are initialized by setting each to be equal to a constant value of one inside the corresponding coarse block, i.e.,

$$\mathcal{P}^0_{i,K} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \Omega^c_K, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Then, we compute the iterative increments

$$\hat{d}_K = -\omega D^{-1} A \mathcal{P}^\eta_K, \tag{17}$$

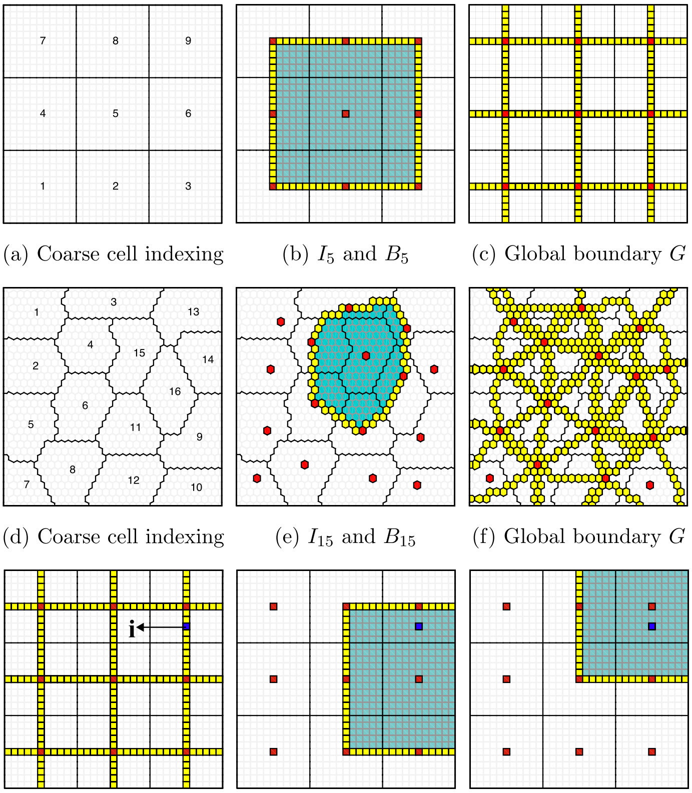
where A is the fine-scale system, $D = diag(A)$ is the diagonal entries of A, and ω is a relaxation (or damping) parameter, which is set to 2/3 for all simulations reported in this paper.

To ensure that basis functions have local support, the increments $\hat{d}_K$ must be restricted to have nonzero values only inside I_K. This is done by setting $\mathcal{P}^\eta_K$ outside the support region to zero and normalizing all other basis functions whose support regions include the boundary cells B_K, i.e.,

$$d_{iK} = \begin{cases} \dfrac{\hat{d}_{iK} - \mathcal{P}^\eta_{iK} \sum_{J \in H_i} \hat{d}_{iJ}}{1 + \sum_{J \in H_i} \hat{d}_{iJ}}, & \text{if } \mathbf{x}_i \in I_K \cap G, \\ \hat{d}_{iK}, & \text{if } \mathbf{x}_i \in I_K \setminus G, \\ 0, & \text{if } \mathbf{x}_i \notin I_K. \end{cases} \tag{18}$$

This modified increment is now used to update the prolongation operator, i.e.,

$$\mathcal{P}^{\eta+1}_K = \mathcal{P}^\eta_K + d_K. \tag{19}$$



(a) Coarse cell indexing (b) I_5 and B_5 (c) Global boundary G

(d) Coarse cell indexing (e) I_{15} and B_{15} (f) Global boundary G

(g) A fine cell $\mathbf{x}_i \in G$ (h) $\mathbf{x}_i \in I_6$ (i) $\mathbf{x}_i \in I_9$

Fig. 1. The top row shows a rectangular grid with a uniform coarse partition, while the second row presents a hexagonal grid with an unstructured coarse partition. The left images (**a** and **d**) show the indexing scheme for the coarse blocks. In the middle images (**b** and **e**), cells inside the support regions for a coarse node are highlighted in turquoise, whereas yellow color signifies the support boundary. The right images (**c** and **f**) show the global boundary cells G highlighted in yellow. The last row uses the same grid as in the first row to depict the set H_i for a particular cell $\mathbf{x}_i \in G$ marked in blue (**g**). The next two images (**h** and **i**) show the support regions that make up the set H_i and contain that particular cell. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To measure convergence of the basis functions, a local error e_K is defined outside G,

$$e_K = \max_i(|\hat{d}_{iK}|), \quad \mathbf{x}_i \notin G, \tag{20}$$

and basis functions are assumed to be converged if $||e||_\infty \leq$ tolerance after any increment. If not, we set $\mathcal{P}_K^{\eta} = \mathcal{P}_K^{\eta+1}$ and repeat the steps (17)–(19).

Note that the basis functions can be constructed using parallel processing. By virtue of the fine-scale discretization scheme used to construct the system matrix A, every successive increment computed using (17) will only spread the corresponding basis functions further by a topological distance of 1. Hence, once the basis function P_K covers its support region I_K, the next increment will spread into its support boundary B_K. Using the third expression in (18), these nonzero values outside the support region would be set to zero. This could lead to the prolongation operator not having a partition of unity. To reimpose partition of unity in the prolongation matrix, the discarded values are redistributed within other basis functions with support in these fine cells. Indices of such support regions are stored in the set H. In essence, the first and the last expression in (18) explicitly impose a partition of unity in all cells belonging to the global boundary G.

Fig. 2 shows how this iterative procedure gradually converges to the standard FEM hat function for a homogeneous 1D medium, while Fig. 3 shows basis functions for three different 2D permeability fields.

4. MsRSB for fractured media (F-MsRSB)

The F-MsRSB method is devised on the basis of introducing basis functions for both matrix and fracture domains, similar to F-AMS [50], i.e., $p \approx p' = [p'_m \quad p'_f]^T$, where

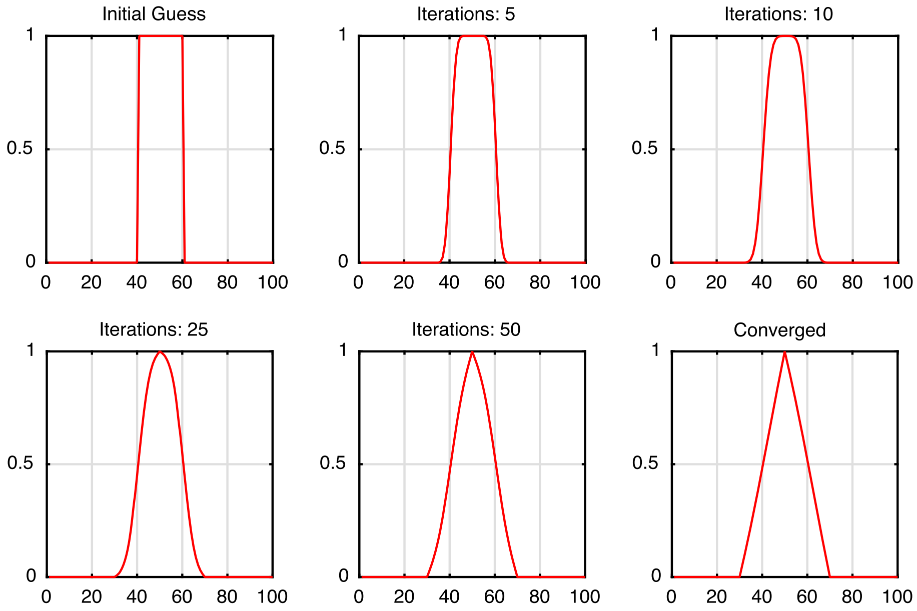


Fig. 2. Restriction smoothed basis function computed iteratively over a 100 m long 1D homogeneous domain with 20 m coarse blocks. The vertical axis gives the basis-function value for the 3rd coarse block. Tolerance for convergence $= 10^{-3}$.

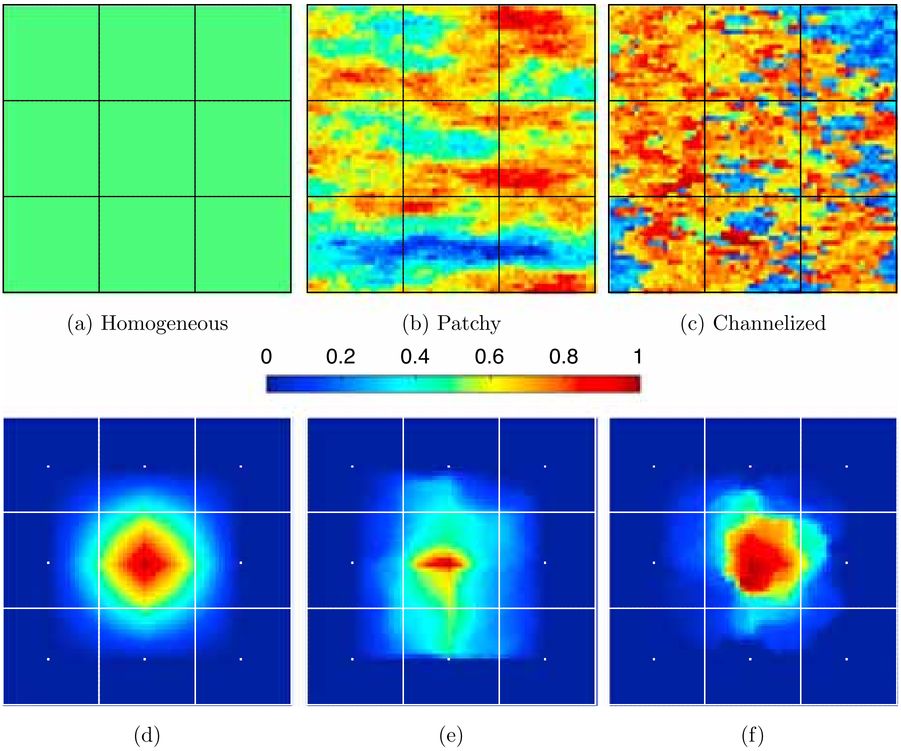


Fig. 3. Illustration of the restriction smoothed basis functions (**d–f**) for three different permeability fields: homogeneous (**a**), heterogeneous patchy-field (**b**), and heterogeneous channelized field (**c**).

$$p'_m = \sum_{j=1}^{n_{cm}} \Phi_j^{m,m} p_j^{c,m} + \sum_{j=1}^{n_{cf}} \Phi_j^{f,m} p_j^{c,f}, \tag{21}$$

$$p'_f = \sum_{j=1}^{n_{cm}} \Phi_j^{m,f} p_j^{c,m} + \sum_{j=1}^{n_{cf}} \Phi_j^{f,f} p_j^{c,f}. \tag{22}$$

Here, p'_m and p'_f are approximate matrix and fracture pressures, respectively, computed by MsRSB at the fine scale. In addition, coarse-scale solutions in matrix and fracture are denoted as $p^{c,m}$ and $p^{c,f}$, respectively. There exist n_{cm} coarse matrix blocks and n_{cf} coarse fracture blocks in total. It is important to realize that there may be several disconnected fracture networks in a reservoir model. Each independent fracture network can contain one or more fracture coarse cells, so that n_{cf} comprises all fracture coarse blocks across all fracture networks. Moreover, $\Phi^{m,m}$ and $\Phi^{m,f}$ are basis functions for matrix coarse blocks with superscripts "m, m" and "m, f" denoting values in the matrix and fracture domains, respectively. Both $\Phi^{f,f}$ and $\Phi^{f,m}$ are fracture basis functions with superscript "f, f" representing the values inside the corresponding fracture network and superscript "f, m" denoting contributions inside the matrix domain. Hence, the prolongation operator can be written as

$$\mathcal{P} = \left[\begin{array}{ccc|ccc} \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m,m} & \cdots & \Phi_{n_{cm}}^{m,m} & \Phi_1^{f,m} & \cdots & \Phi_{n_{cf}}^{f,m} \\ \vdots & & \vdots & \vdots & & \vdots \\ \hline \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{m,f} & \cdots & \Phi_{n_{cm}}^{m,f} & \Phi_1^{f,f} & \cdots & \Phi_{n_{cf}}^{f,f} \\ \vdots & & \vdots & \vdots & & \vdots \end{array}\right]_{n_f \times n_c}, \tag{23}$$

where $n_f = (n_f^m + n_f^f)$ and $n_c = (n_{cm} + n_{cf})$ are total degrees-of-freedom (matrix and fractures) at fine and coarse scales, respectively.

Generally, fractures are much more conductive than the matrix rock. Full consideration of both fracture and matrix coarse solutions, $p^{c,m}$ and $p^{c,f}$, for interpolated fracture pressure, p'_f, can lead to improved convergence properties. However, such an approach results in much denser prolongation operators. Therefore, the improvement in convergence rate may not necessarily offset the additional computational cost. Numerical studies of F-AMS for 3D problems (considering CPU time), support the idea of eliminating the effect of matrix coarse pressure in the fracture pressure interpolation, i.e., setting $\Phi^{m,f} = 0$. In this paper, the same sparse operator is considered.

Next, the support region and the procedure for calculating basis functions for fractured media are explained.

4.1. Support regions and basis functions

The support region for each fracture block is generated based on a topological distance. More precisely, the support region for a fracture block includes all fine cells located inside the sphere (circle in 2D) with radius d in index space. Note that d is an integer input to the simulator but there is no algorithmic restriction on using expressions to automatically compute d. Optimization of the choice of fracture support regions would require a more detailed study, considering both accuracy and efficiency, and also taking into account the effective coarsening ratio used to compute the multiscale solution. Such a study is beyond the scope of this paper and subject of a future study. It is clear that the support region will include no fine-cell except those overlapping with the fractures, if $d = 0$ is considered. For the numerical examples studies in this paper the value of $d = 7$ is used. This is of the same order as the matrix coarsening ratio used in our examples (i.e. 10 in each direction). An overview of the procedure to generate the fracture support region is presented in Algorithm 1.

Fig. 4 illustrates coarse grids and support regions inside which basis functions are compactly supported for a test case with 30 × 30 matrix and 20 fracture cells. Furthermore, Fig. 5 shows basis functions inside the matrix rock (belonging to both fracture and matrix coarse nodes).

Algorithm 1 Generating fracture support regions.

```
Initialize: $A =$ Adjacency matrix for the fine-scale system, $d \in \mathbb{Z}$ and $m = 1$
for $J \in \{1, ..., n_{cf}\}$ do
    $I_{i,J} = 1$ if $\mathbf{x}_i \in \Omega^c_J$,   $I_{i,J} = 0$ otherwise
    while $m < d$ do
        $I_J = A \times I_J$
        $m = m + 1$
    end while
end for
```

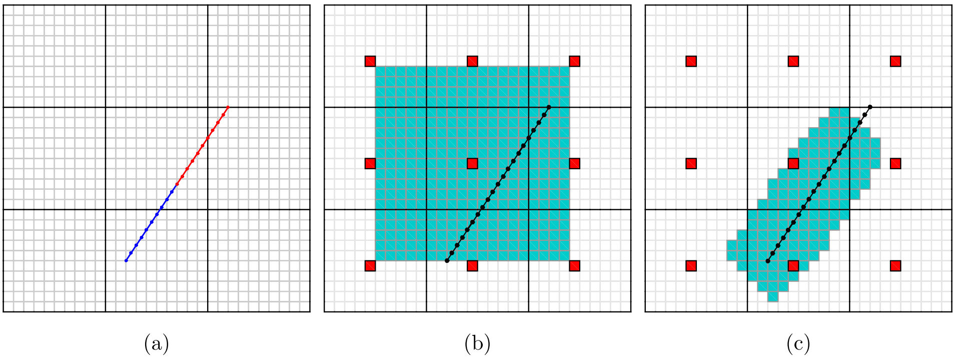
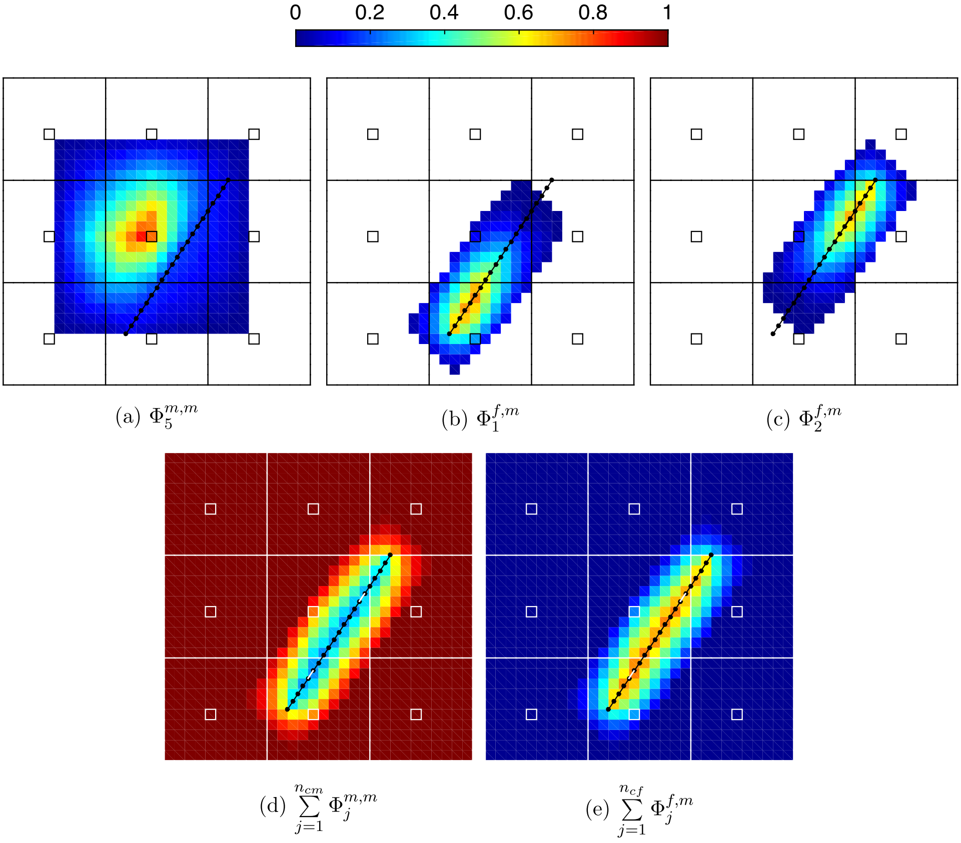


Fig. 4. Illustration of multiscale grids for matrix and fracture **(a)** with support regions for matrix **(b)** and fracture **(c)** for a case with 30 × 30 matrix and 20 fracture fine-scale cells. Multiscale coarse grid contains 3 × 3 matrix blocks and 2 blocks inside the fracture. Here, $d = 7$.



(a) $\Phi_5^{m,m}$ (b) $\Phi_1^{f,m}$ (c) $\Phi_2^{f,m}$

(d) $\sum_{j=1}^{n_{cm}} \Phi_j^{m,m}$ (e) $\sum_{j=1}^{n_{cf}} \Phi_j^{f,m}$

Fig. 5. Matrix and fracture basis functions for the system described in Fig. 4.

5. Numerical results

The developed F-MsRSB method is implemented and integrated with the free, open-source MATLAB Reservoir Simulation Toolbox (MRST) [61,67,68,60]. In this section, we investigate the performance of F-MsRSB for many challenging cases. The numerical examples involve both 2D and 3D heterogeneous media. Next, the sensitivity of the method to coarse-grid resolution for fracture domain is studied. Then, its performance for heterogeneous rock formations is studied through a realistic fracture map obtained from an outcrop and for a statistically generated fracture map. Using an outcrop map is unique in the literature of multiscale methods for fractured media. For the statistical map, we use METIS [57] to generate an unstructured

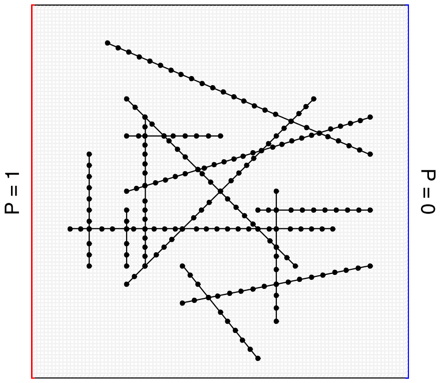
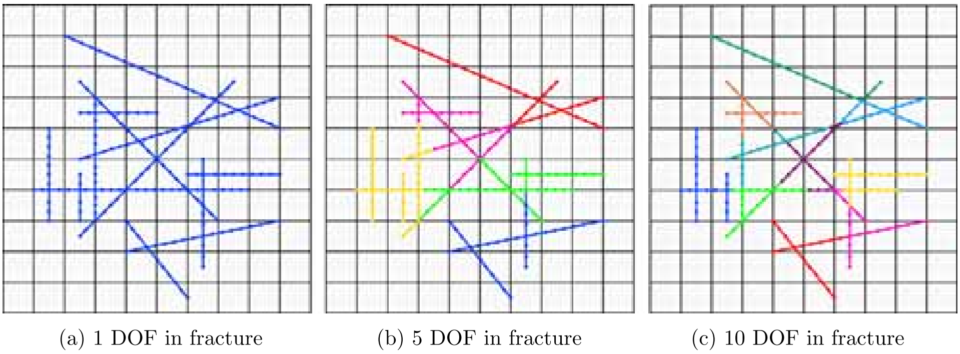


Fig. 6. First test case which contains 100 × 100 matrix and 200 fracture cells at the fine scale, with homogeneous 2D rock formation. The left and right boundaries are subject to Dirichlet values of 1 and 0, respectively.


(a) 1 DOF in fracture

(b) 5 DOF in fracture

(c) 10 DOF in fracture

Fig. 7. F-MsRSB coarse grids for matrix and fractures. Each coarse matrix block consists of 10 × 10 fine cells, while the fracture blocks are varied from 1 (200 fine cells in 1 block) to 5 (40 cells in each block) and 10 (20 cells in each block).

partition of a fine-scale hexagonal grid representing the matrix rock. Finally, three test cases with 3D heterogeneous matrix properties along with 2D fracture plates are considered to provide the scientific community with a reliable assessment of the devised F-MsRSB method.

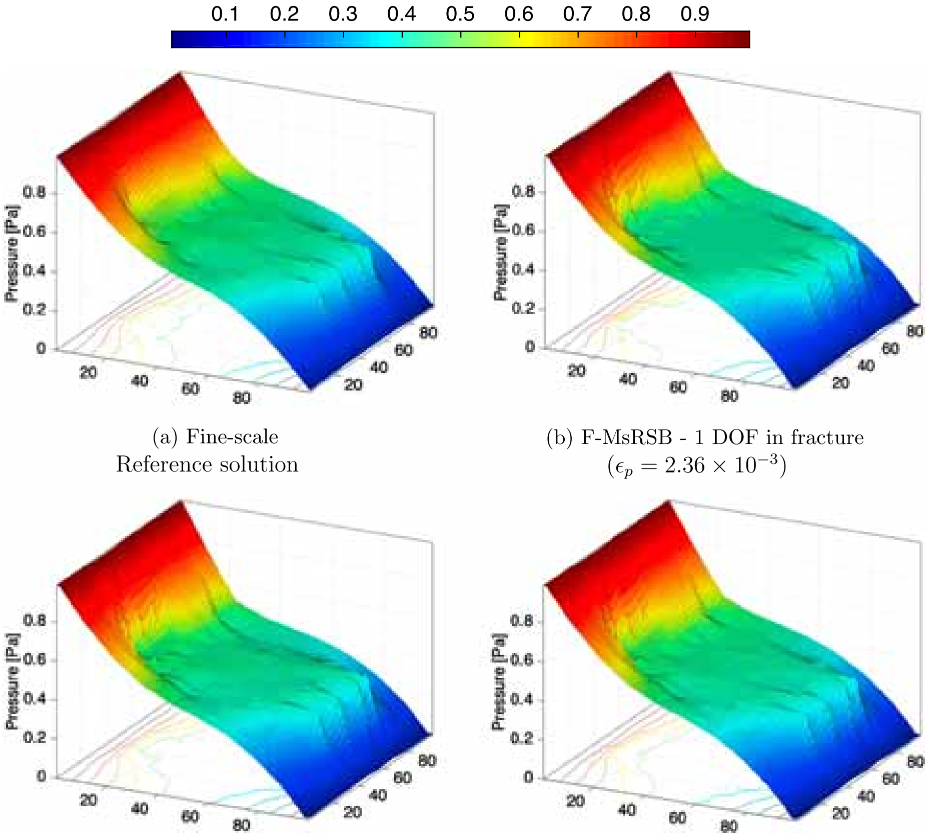
5.1. Sensitivity to coarse DOFs in fracture

In this test case, we consider single-phase flow in a 100 × 100 m^2 homogeneous domain to study the effect of the coarsening ratio in the fracture domain on the accuracy of the multiscale method as well as its preconditioning properties. The matrix permeability is set to 1 Darcy and $k^f/k^m = 10^4$. Fluid viscosity is 1 cP. The matrix contains 100 × 100 grid cells. As shown in Fig. 6, the matrix contains one fracture network described using 200 cells. A coarsening ratio of 10 × 10 is fixed for the matrix domain, while the coarsening ratio for the fracture is varied from 1 to 50 (some cases are shown in Fig. 7). By increasing the number of coarse degrees-of-freedom in the fracture, the F-MsRSB pressure solution improves (Fig. 8). This finding is consistent with that of F-AMS [50].

Fig. 8 shows pressure solutions after one multiscale cycle for different fracture coarsening ratios. As a quantitative error measurement, the F-MsRSB pressure error is calculated using a scaled discrete L^2 norm

$$\epsilon_p = \sqrt{\frac{\sum_{i \in n_f} (p_i^{fs} - p_i^{ms})^2 |\Omega_i|}{\sum_{i \in n_f} (p_i^{fs})^2 |\Omega_i|}}. \tag{24}$$

Errors for different grid sizes inside the fracture are also provided in Fig. 8. In Fig. 9, we have used Pollock's method [70] to trace streamlines for the fine-scale reference solution and the multiscale solution computed using different degrees of freedom in the fracture. This provides an excellent way to visualize the flow field and study the accuracy of the multiscale velocity profile. Additionally, to quantify the accuracy of the velocity field we consider a vertical slice in the middle of the matrix domain, as shown in Fig. 10a. The total flow rates obtained by F-MsRSB (q^{ms}) and the fine-scale discretization (q^{fs}) are compared across this section. Fig. 10b reports the discrepancy



(a) Fine-scale Reference solution

(b) F-MsRSB - 1 DOF in fracture ($\epsilon_p = 2.36 \times 10^{-3}$)

(c) F-MsRSB - 5 DOF in fracture ($\epsilon_p = 2.02 \times 10^{-3}$)

(d) F-MsRSB - 10 DOF in fracture ($\epsilon_p = 1.67 \times 10^{-3}$)

Fig. 8. Reference and F-MsRSB pressure for the first test case as shown in Fig. 6. F-MsRSB solutions are presented for different fracture coarsening ratios.

$$\epsilon_q = 100 \times \frac{|q^{fs} - q^{ms}|}{|q^{fs}|} \tag{25}$$

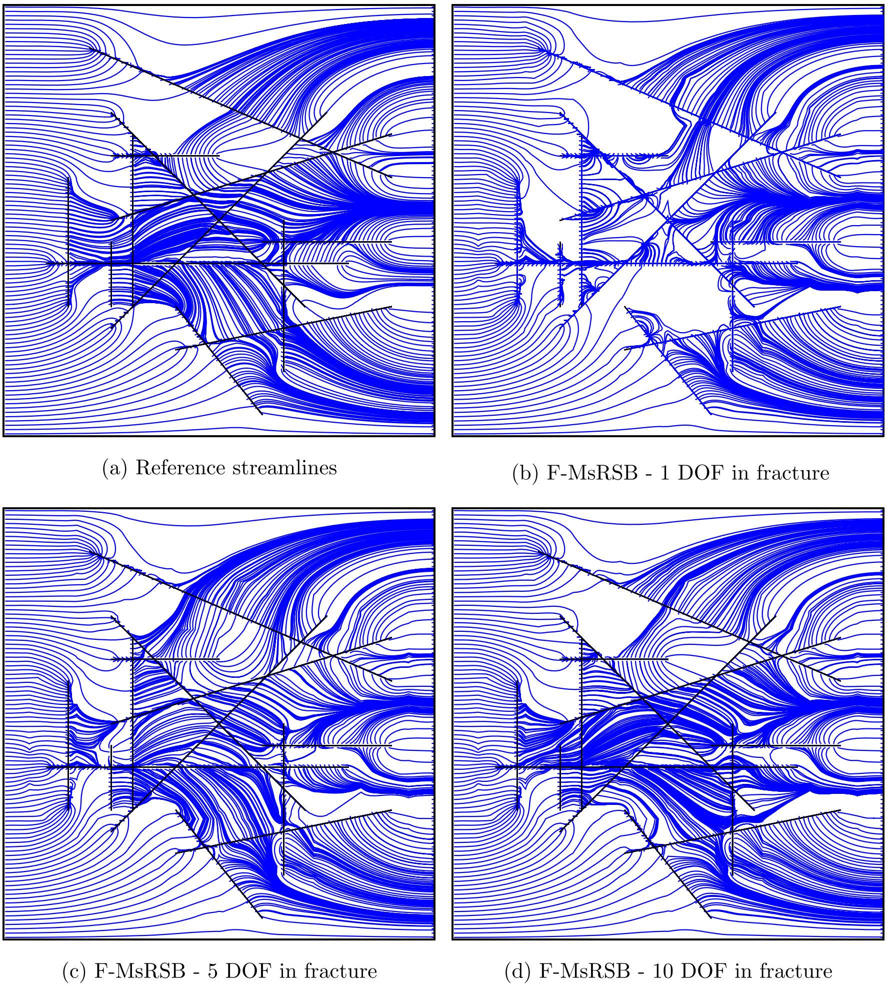
at this cross section as a function of fracture coarsening ratio. Similarly to F-AMS [50], we observe that the condition number of the F-MsRSB coarse system improves by increasing the coarse resolution inside the fracture. This leads to higher iterative convergence rates, as shown in Fig. 11, if F-MsRSB is combined with ILU(0) in an iterative multiscale procedure [50]. Convergence is determined on the basis of setting a threshold value for the scaled residual norm, i.e., $\|r_b\|_2 = \|Ap - q\|_2 / \|q\|_2$.

5.2. F-MsRSB for heterogeneous fractured media

To study F-MsRSB for heterogeneous fractured media, two fracture maps are considered: (i) the fracture map is extracted from an outcrop of dimensions 246.3 × 283.1 m^2; (ii) a statistical fracture model is generated over an unstructured PEBI grid. The heterogeneous rock property is assumed to represent heterogeneity variations in the matrix rock along with homogenized small-scale fractures.

5.2.1. Outcrop fracture map

The fracture coordinates are scaled from an outcrop photo [69] to fit a domain size of 1000 × 1000 m^2, as shown in Fig. 12a. The fine-scale grid contains 100 × 100 matrix and 2074 fracture cells (over 94 disconnected fracture networks), while the F-MsRSB grid contains 15 × 15 matrix and 155 fracture blocks. Fig. 12b shows the permeability of the matrix formation. Fracture permeability is set to 1000 Darcy.


(a) Reference streamlines

(b) F-MsRSB - 1 DOF in fracture

(c) F-MsRSB - 5 DOF in fracture

(d) F-MsRSB - 10 DOF in fracture

Fig. 9. Reference and F-MsRSB velocity field represented by streamlines for the first test case as shown in Fig. 6. F-MsRSB solutions are presented for different fracture coarsening ratios.

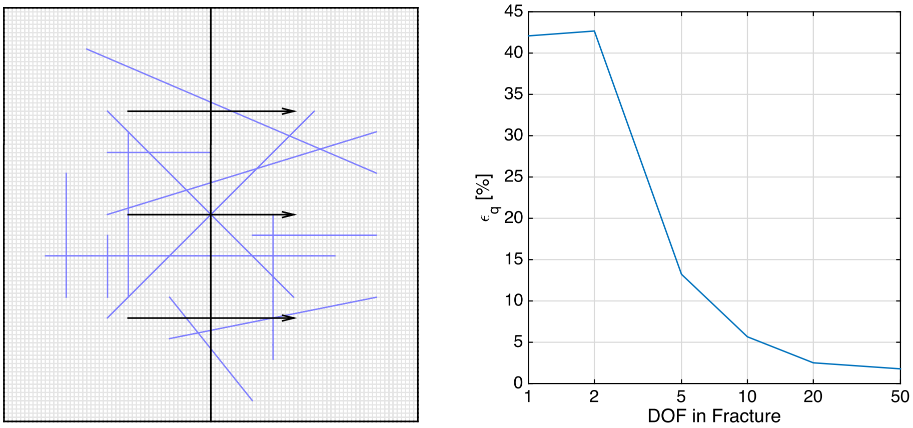
We consider two-phase flow, with quadratic relative permeability curves and unit viscosity ratio between the phases. Fluid is injected at a constant rate in cell $(1, 100)$, while fluids are produced in the opposite corner $(100, 1)$ at constant pressure. Fig. 13 shows the saturation maps obtained after one F-MsRSB cycle (no iterations) compared with reference fine-scale solutions, both obtained using a sequentially-implicit strategy. Clearly, the higher the resolution of fracture coarse grid, the more accurate the F-MsRSB results.

Saturation error is calculated as

$$\epsilon_S = \frac{\max_{i\in n_f}\left(|S_i^{fs} - S_i^{ms}||\Omega_i|\phi_i\right)}{\max_{i\in n_f}\left(|S_i^{fs}||\Omega_i|\phi_i\right)}, \tag{26}$$

which, because it is scaled with pore volume, gives a very strict measure of the error in the spatial mass distribution for incompressible fluids.

Clearly, multiscale solutions can be improved by increasing the number of degrees-of-freedom per fracture network or by applying iterations (in combination with ILU(0), similar to [50,58,47,53]). Fig. 14 shows overall saturation errors. After only a few iterations of the two stage (F-MsRSB + ILU(0)) cycle, the multiscale fluid distribution is virtually identical to the reference solution. Convergence to a tolerance of 0.1 takes 8 iterations for this outcrop model, whereas a tolerance of 0.01 is reached after 20 iterations. One can also employ a local block solver around the fractures and wells [9].



(a) Vertical slice through the center of the domain

(b) % Error in flow rate through a vertical slice at the center of the domain

Fig. 10. Vertical cross section through the center of the matrix domain (**a**). Percentage error in total flow rate through this vertical cross section in the matrix (**b**).

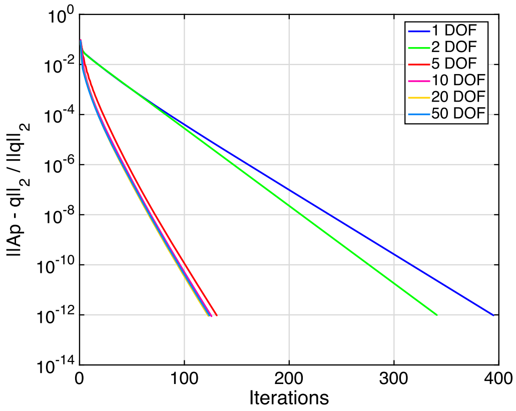
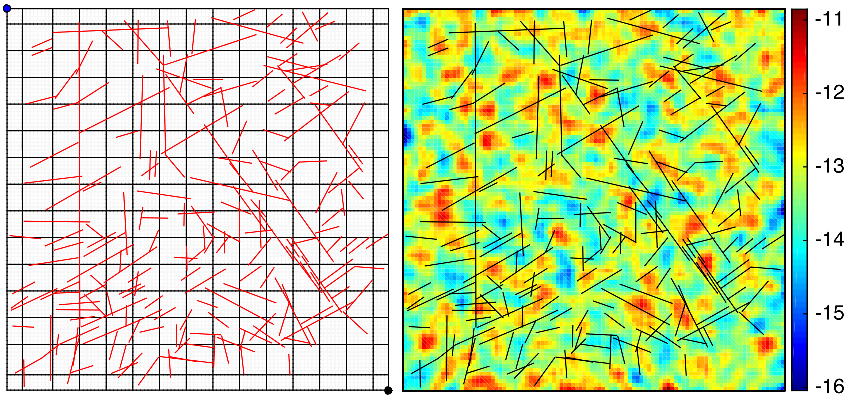


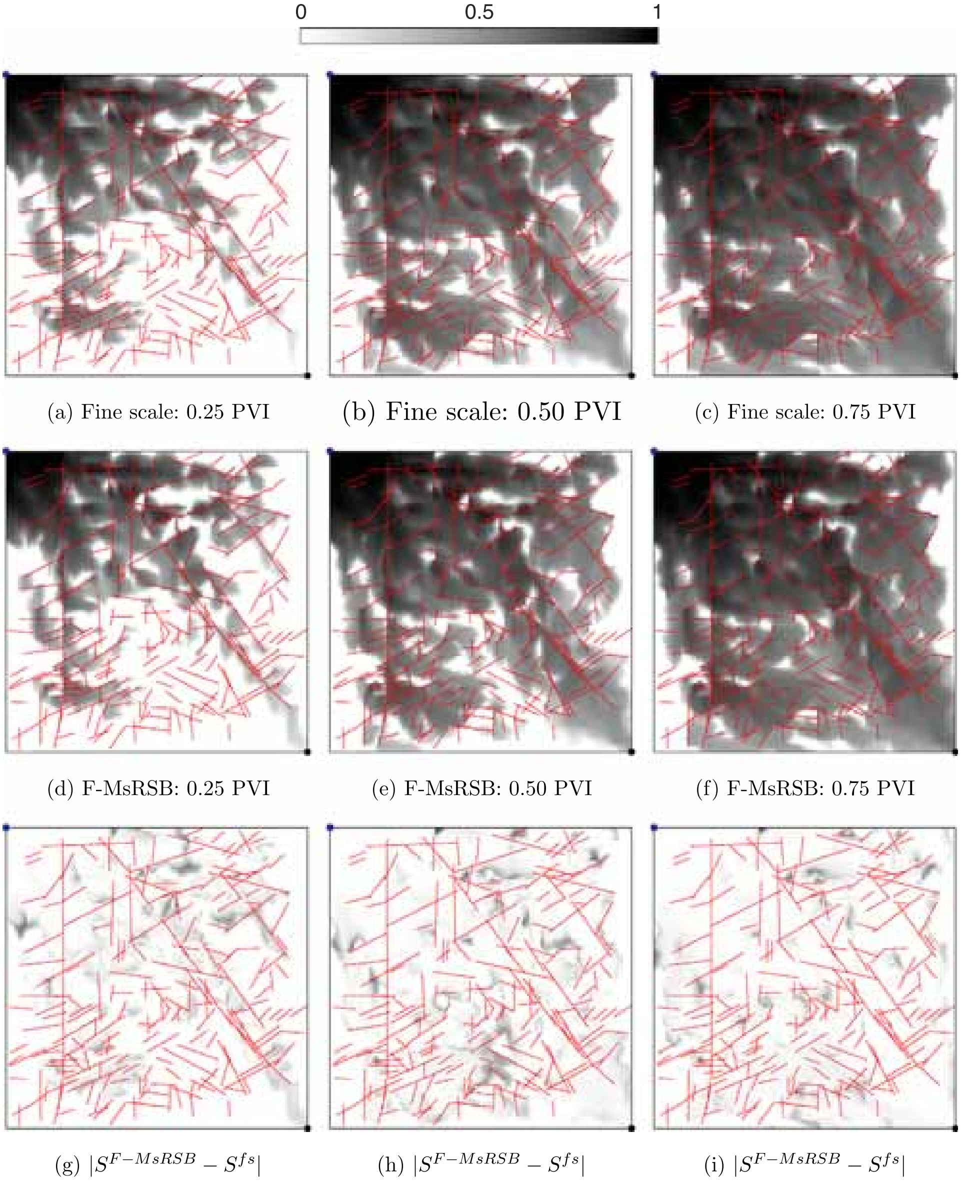
Fig. 11. Convergence of F-MsRSB + ILU(0) for different DOF in fracture.



(a) Coarse partition and fractures

(b) $\log_{10}(k^m)$ [m^2]

Fig. 12. Coarse grid in matrix showing well locations and fracture map extracted from an outcrop (**a**). Logarithm of the corresponding permeability field in the matrix (**b**).



(a) Fine scale: 0.25 PVI

(b) Fine scale: 0.50 PVI

(c) Fine scale: 0.75 PVI

(d) F-MsRSB: 0.25 PVI

(e) F-MsRSB: 0.50 PVI

(f) F-MsRSB: 0.75 PVI

(g) $|S^{F-MsRSB} - S^{fs}|$

(h) $|S^{F-MsRSB} - S^{fs}|$

(i) $|S^{F-MsRSB} - S^{fs}|$

Fig. 13. Reference saturation profile compared with saturation maps obtained after one F-MsRSB cycle at different PV injected. Absolute errors in saturation are also shown in (**g**)–(**i**).

5.2.2. Statistical fracture model

A 1000 × 500 m^2 heterogeneous domain with permeability and porosity sampled from the 10th layer of the SPE10 dataset [56] is considered and shown in Fig. 15. Similar as in previous test case, fracture permeability and porosity are 1000 Darcy and 0.50, respectively. PEBI grids are employed for the matrix at fine scale, which is a Voronoi map over a uniform triangulation in the region. The fine-scale grid contains 4726 cells for matrix and 2207 cells for fractures. There exist 55 disconnected fracture networks in the domain. Both matrix and fracture are coarsened using METIS [57] to give 100 blocks for each domain (fracture and matrix) as shown in Fig. 15a.

Two incompressible fluid phases with quadratic relative permeabilities are considered. The reservoir is initially filled with oil having a viscosity of 5 cP. Water with a lower viscosity of 1 cP is injected at a constant rate from a well near the bottom-left corner, while a the producing well near the top right corner of the domain is kept at constant pressure. Fig. 16 shows the saturation maps after one multiscale cycle (F-MsRSB + ILU(0)) for injection amounts of 0.2, 1.0, and 1.8 pore

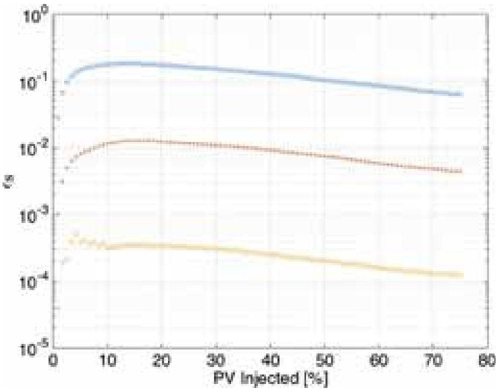
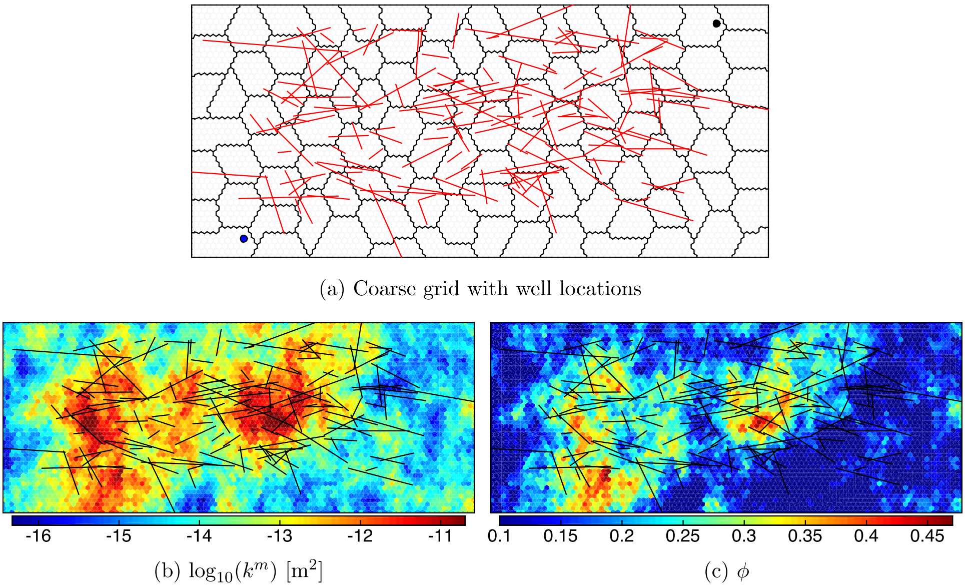


Fig. 14. Saturation error for the outcrop model as a function of simulation time measured in pore-volume-injection. Shown are the non-iterative and the iterative F-MsRSB method with tolerances of 0.1 and 0.01 on the pressure solves.



(a) Coarse grid with well locations

(b) $\log_{10}(k^m)$ [m^2]

(c) ϕ

Fig. 15. Matrix coarse grid with 100 DOF, fracture map, and well locations (**a**). Petrophysical rock properties sampled from the 10th layer of the Tarbert formation in the SPE10 dataset (**b** and **c**).

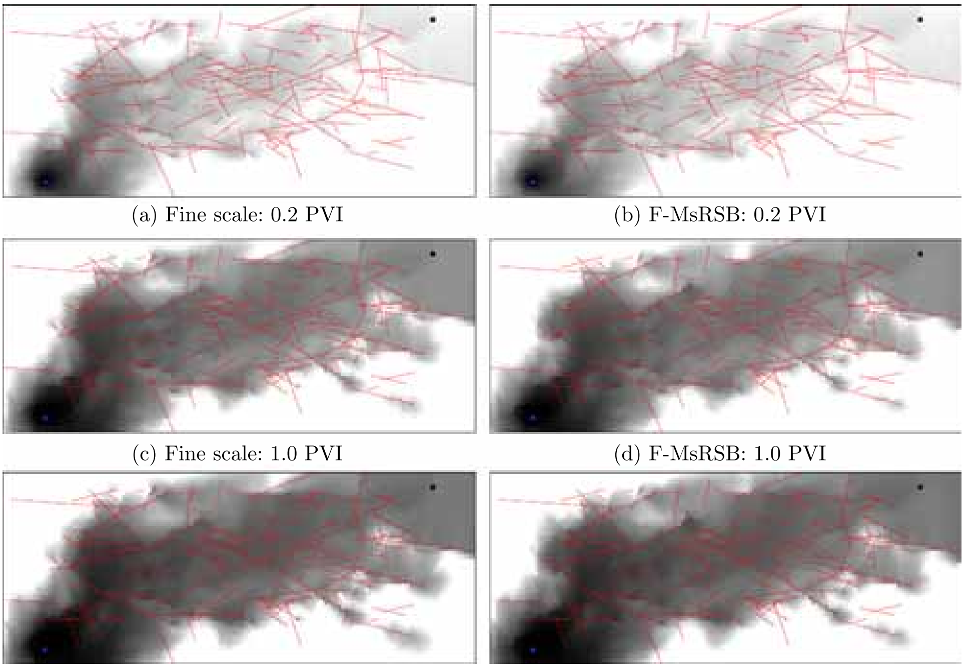
volumes (PVI). The initial multiscale solution is already quite accurate, and after one smoothing-iteration step, the multiscale and reference solutions are virtually identical. Fig. 17 compares bottom-hole pressure in the injector oil rate in the producer as computed by the fine-scale reference solver and F-MsRSB with different iterative tolerances.

5.3. 3D models

In this section, we study the performance of F-MsRSB for three examples in which the matrix domain is described in 3D and fractures are planar 2D surfaces.

5.3.1. Two intersecting fracture planes

The fine-scale grid for the first 3D example consists of 50 × 50 × 50 matrix cells and two fracture planes, each with 100 × 30 fracture grid cells. The fracture planes cross in the middle of the domain, as shown in Fig. 18a. Fluid is injected at a constant rate at the bottom-left corner, while production takes place from the top-right corner. Matrix permeability is shown in Fig. 18b. The F-MsRSB grid contains 20 × 20 × 20 matrix blocks, whereas each fracture plan is partitioned into 12 × 4 fracture blocks. Fracture permeability is set to 10^4 Darcy. The matrix coarsening ratio is chosen such that the effect of fracture coarsening ratios will be more pronounced in the F-MsRSB results. Fig. 19 shows the pressure solution



(a) Fine scale: 0.2 PVI

(b) F-MsRSB: 0.2 PVI

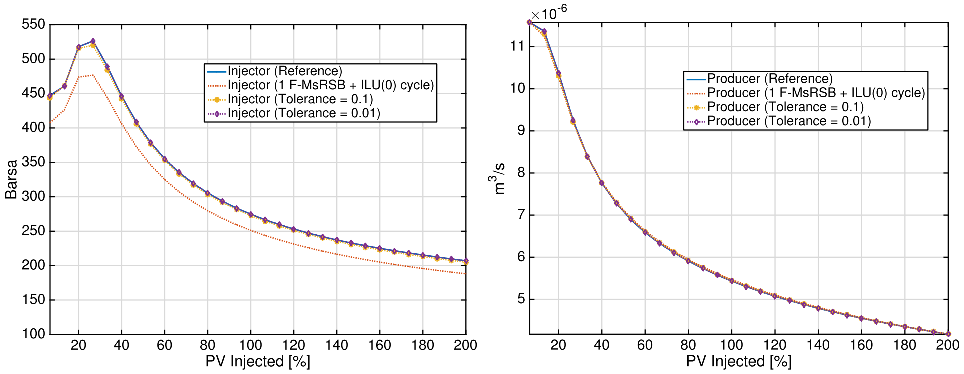
(c) Fine scale: 1.0 PVI

(d) F-MsRSB: 1.0 PVI

(e) Fine scale: 1.8 PVI

(f) F-MsRSB: 1.8 PVI

Fig. 16. Saturation maps at different times computed by the fine-scale reference solver and F-MsRSB with one iteration cycle (F-MsRSB + ILU(0)).



(a) Injector bottom hole pressure

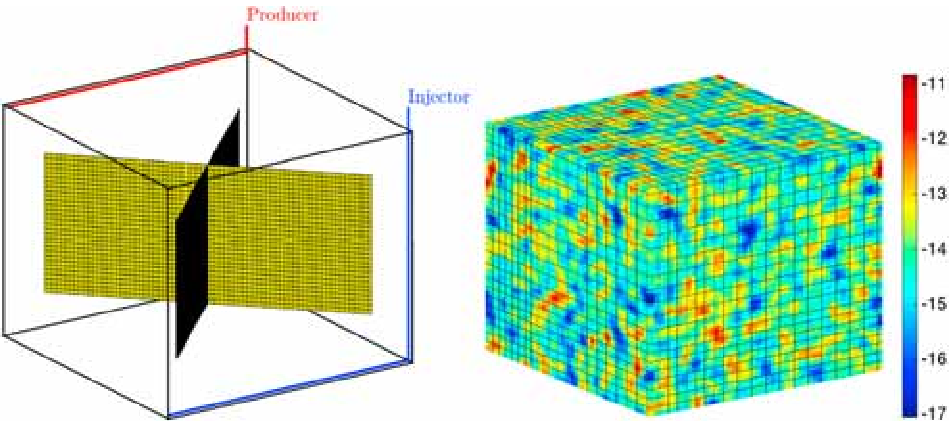
(b) Oil rate at producer

Fig. 17. Production and injection quantities at well locations for the statistical 2D fracture map.

obtained after one F-MsRSB step. In addition, Fig. 20 presents the convergence behavior for different coarse resolutions for the fracture planes. We observe that a modest increase in the fracture resolution, from having a single DOF for each plane, leads to significantly improved convergence rates.

5.3.2. *Depositional bed*

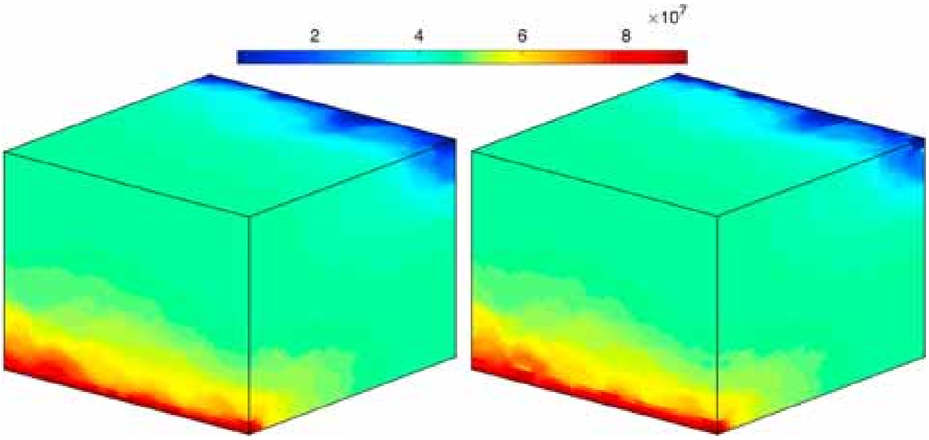
As another 3D example, we consider a corner-point grid modeling a core-scale depositional bed. A similar model has been used in the literature [8] as an example of a model with a large number of thin, low-permeable shale layers pinched between layers of good sands. Pinch-outs, owing mainly to erosion, are a common reason behind unstructured cell connections in stratigraphic corner-point grids. They lead to degenerate cells with faces of zero area resulting in a complex grid geometry. With the addition of heterogeneity, it becomes quite a challenging test case for multiscale finite-volume methods [11,18]. We make the model even more complicated by adding inclined fracture planes in the interior of the model. To im-



(a) Fracture planes and wells.

(b) $\log_{10}(k^m)$ [m^2] and coarse partition.

Fig. 18. Illustration of the first 3D test case, with a fine-scale grid that contains 50 × 50 × 50 matrix cells and two intersecting fracture planes that each contains 100 × 30 fracture cells. Also shown on the right is heterogeneous matrix permeability map, along with the imposed 20 × 20 × 20 coarse grid used by F-MsRSB.



(a) Reference solution

(b) Multiscale solution

Fig. 19. Reference and multiscale pressure solution after one F-MsRSB cycle for single-phase flow in the simple 3D model. Each fracture plane is logically partitioned into 12 × 4 blocks.

prove the efficiency of the F-MsRSB preprocessing steps for this challenging grid geometry, we first calculate the C_I factors globally for each fracture plane and then for each fracture–matrix overlapping discrete grid cell.

The fine-scale grid contains 30 × 30 × 100 matrix with 222 hexagonal fracture cells per fracture plane. As shown in Fig. 21a, there exist 6 fracture planes, and the domain is subject to Dirichlet boundary conditions on the left and right faces, while all other faces are subject to no-flow condition. As shown in Fig. 21, F-MsRSB employs 10 × 10 × 9 blocks for the matrix rock, and only 2 blocks for each fracture plane. In addition, fracture aperture is 0.04 m. The matrix permeability distribution is shown in Fig. 21b, and we set $k_f = 10^4$ D, resulting in large contrasts in the permeability values throughout the entire model.

Fig. 22 shows pressure solutions obtained after one F-MsRSB step. It is clear that F-MsRSB and the fine-scale reference solutions are in good agreement, even with such a large coarsening ratio for the fractures. The absolute difference between the two solutions is depicted in Fig. 22c, with the pressure error (ϵ_p) being 8.79×10^{-4}.

5.3.3. Model 2 of SPE10 with fracture networks

As the final test case in this section, we extract a 30 × 110 × 40 subsample from the full 3D model of the challenging SPE10 data set [56]. As shown in Fig. 23, complex fracture planes (located between layers 11 through 30) are obtained

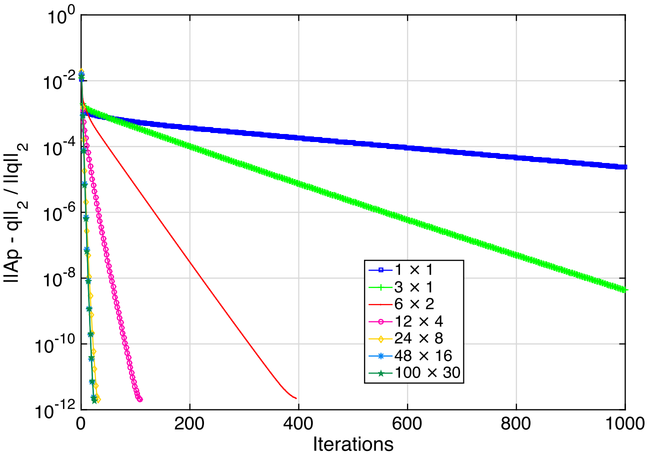
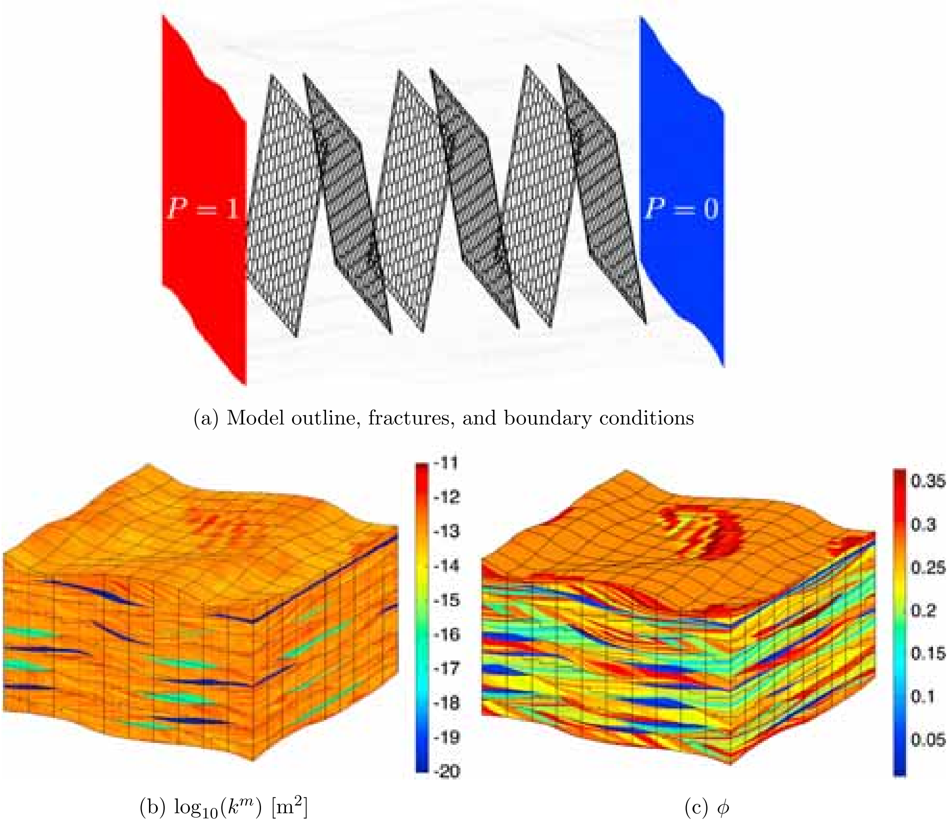


Fig. 20. Convergence property of the F-MsRSB + ILU(0) solver for various coarse resolutions of the two intersecting fracture planes, which each has 100 × 30 cells at the fine scale.



(a) Model outline, fractures, and boundary conditions

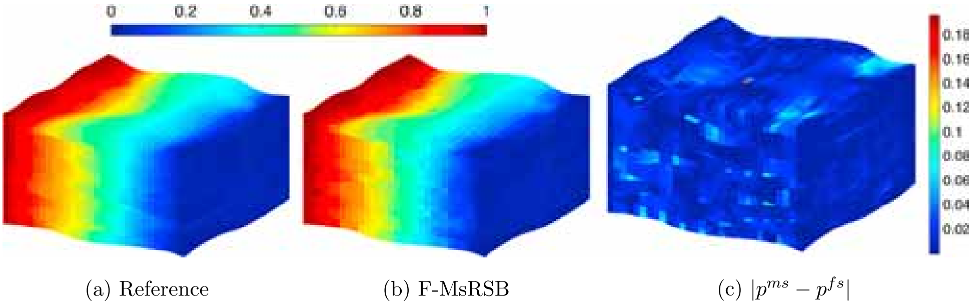
(b) $\log_{10}(k^m)$ [m^2]

(c) ϕ

Fig. 21. Matrix grid with fracture planes and boundary conditions (**a**). Logarithm of permeability map (**b**) and matrix porosity (**c**).

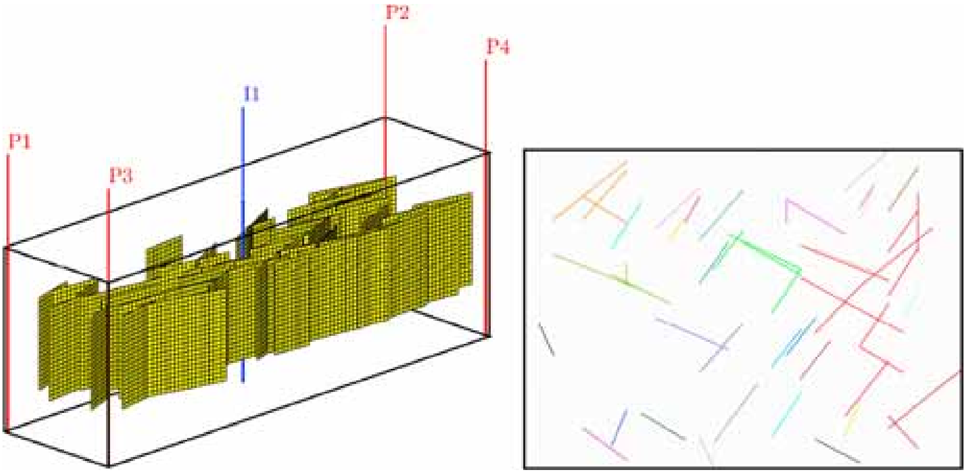
by extruding statistical maps, similar to the ones used for one of the 2D test cases. The model contains 31 disconnected fracture networks, which are discretized using 13, 880 fine-scale grid cells.

Fig. 24 shows the matrix rock properties (permeability and porosity). All fractures have permeability value of 1000 Darcy. The coarse partitions used by F-MsRSB contain 6 × 22 × 8 matrix and 181 fracture blocks in total. Each fracture block contains 80 fine-scale fracture cells. A waterflood experiment has been considered for the duration of 5 PVI, using quadratic



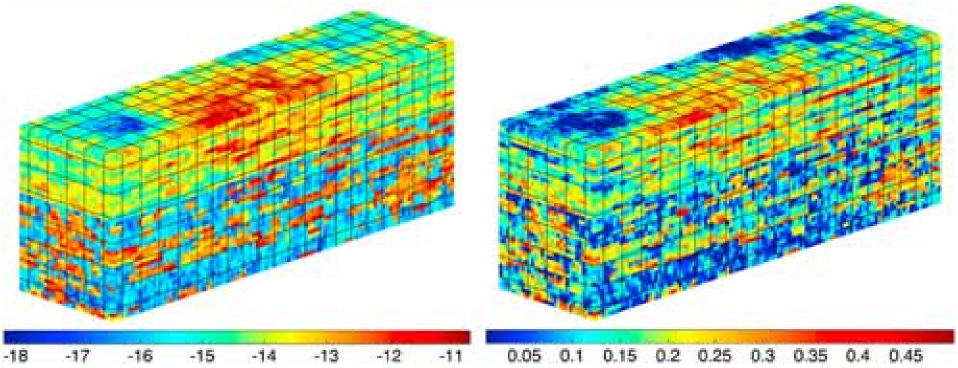
(a) Reference (b) F-MsRSB (c) $|p^{ms} - p^{fs}|$

Fig. 22. Reference and multiscale pressure solution for single-phase flow in the bed model with two degrees-of-freedom per fracture plane.



(a) 3D Test Case 3 (b) Scaled top view

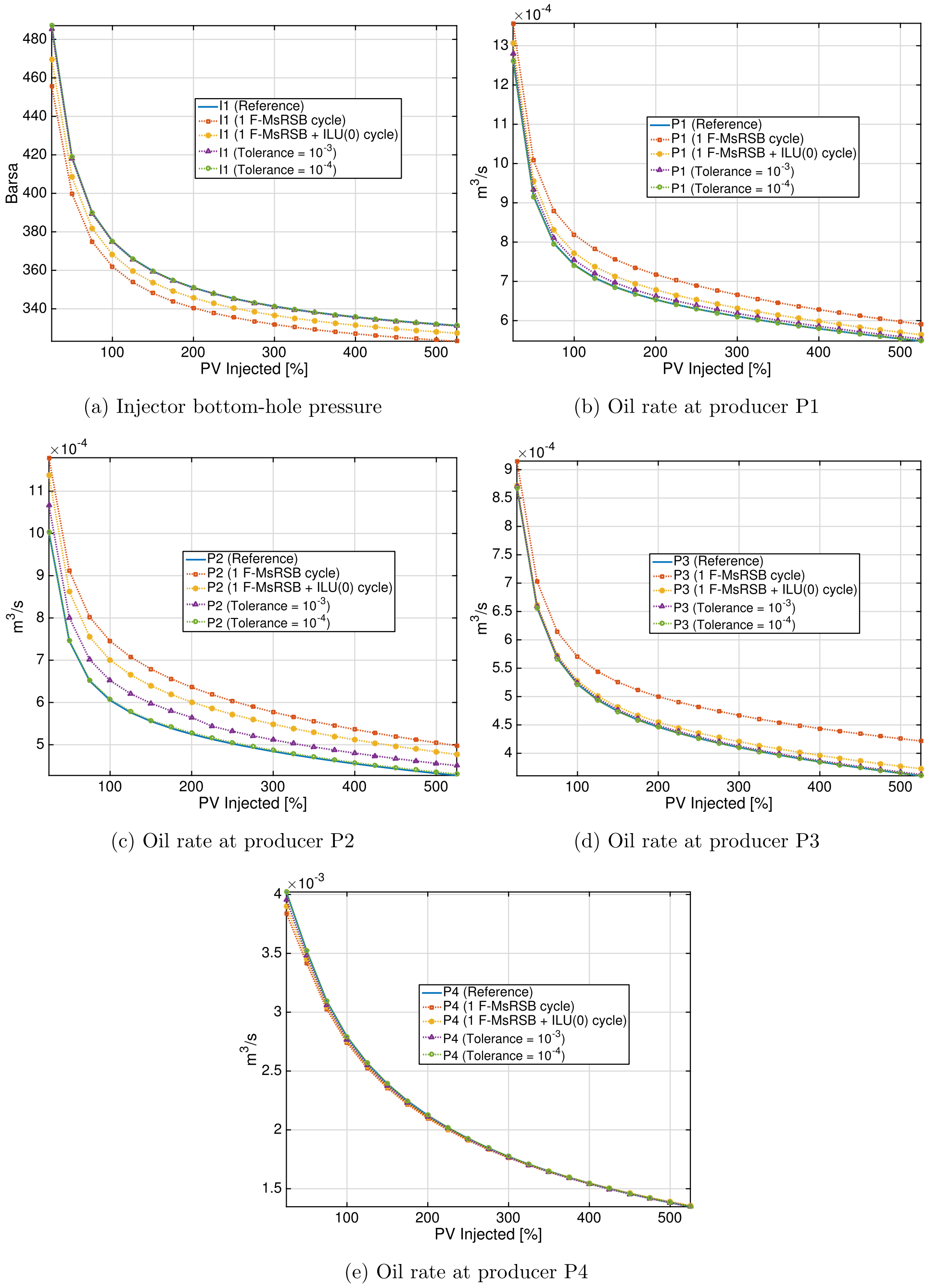
Fig. 23. Five-spot well locations in a subsample of the 3D SPE10 model with 31 disconnected fracture networks added. Shown on the right is the top view of the model.



(a) $\log_{10}(k^m)$ [m^2] (b) ϕ

Fig. 24. Petrophysical properties for the 30 × 110 × 40 domain sampled from the full SPE10 dataset.

relative permeability values. Water with viscosity 1 cP is injected into the reservoir, which is initially filled with 100% oil. Oil viscosity is 10 cP. As shown in Fig. 23a, five wells are placed in a five-spot pattern with a fixed rate injector in the middle and four fixed-pressure producers at the corners.



(a) Injector bottom-hole pressure

(b) Oil rate at producer P1

(c) Oil rate at producer P2

(d) Oil rate at producer P3

(e) Oil rate at producer P4

Fig. 25. Well responses for all wells of the 3D SPE10 test case. F-MsRSB solutions are shown for different tolerances of pressure solution, compared with a fine-scale reference solution.

Well responses computed by F-MsRSB are presented in Fig. 25 and compared with a fine-scale reference solution. As shown, one iteration of (F-MsRSB + ILU(0)) leads to significantly improved solutions. The initial multiscale solution residual is approximately 0.1, which reduces to approximately 0.01 after only one smoothing iteration. Convergence to a tolerance of 10^{-3} and subsequently to 10^{-4} takes approximately 5 and 15 iterations on average, respectively, per time step.

6. Conclusion

In this paper, a novel multiscale framework for fractured porous media (F-MsRSB) was introduced. The method benefited from the most recent developments within the multiscale community, namely F-AMS and MsRSB, and yet, devised a novel approach for robust and efficient treatments of fractured media. Efficiency and accuracy of the devised multiscale method was analyzed for many challenging test cases, including a realistic fracture map from an outcrop. These extensive studies are quite unique in the multiscale community. F-MsRSB is formulated and implemented in an algebraic form using the open-source MATLAB Reservoir Simulation Toolbox (MRST). The codes necessary to run the type of experiments reported herein are thus open to the scientific community; another important contribution of this work.

Through a set of single- and multiphase test cases it was found that MsRSB can accurately simulate models of fractured porous media with highly heterogeneous coefficients and produce approximate solutions with a prescribed fine-scale residual accuracy. The numerical test cases also included complex wells. By using an adaptive iterative strategy, one can trade accuracy for computational efficiency, and still produce mass-conservative, approximate solutions on the fine scale.

The basis functions used in this method can be adaptively updated using efficient global smoothing strategies to account for compressibility and gravity among other physical effects. Ongoing research includes consideration of more challenging fluid and rock physics, along with integration of F-MsRSB into an in-house C++ simulator for comparisons of CPU efficiency.

Acknowledgements

This work was sponsored in part by the Chevron/Schlumberger INTERSECT Technology Alliance and Schlumberger Petroleum Services CV, and in part by Schlumberger Information Solutions and the Research Council of Norway under grant no. 226035. Matei Tene was sponsored by PI/ADNOC.

References

[1] K. Aziz, A. Settari, Petroleum Reservoir Simulation, Blitzprint Ltd., Calgary, Alberta, 1979.
[2] 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.
[3] 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.
[4] P. Jenny, S.H. Lee, H.A. Tchelepi, Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media, J. Comput. Phys. 217 (2006) 627–641.
[5] Y. Efendiev, T.Y. Hou, Multiscale Finite Element Methods: Theory and Applications, Springer, 2009.
[6] V. Kippe, J.E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Comput. Geosci. 12 (2008) 377–398.
[7] J.E. Aarnes, V. Kippe, K.-A. Lie, Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels, Adv. Water Resour. 28 (2005) 257–271.
[8] J.E. Aarnes, S. Krogstad, K.-A. Lie, Multiscale mixed/mimetic methods on corner-point grids, Comput. Geosci. 12 (2008) 297–315.
[9] H. Hajibeygi, P. Jenny, Adaptive iterative multiscale finite volume method, J. Comput. Phys. 230 (2011) 628–643.
[10] I. Lunati, P. Jenny, Multiscale finite-volume method for density-driven flow in porous media, Comput. Geosci. 12 (2008) 337–350.
[11] O. Møyner, K.-A. Lie, A multiscale two-point flux-approximation method, J. Comput. Phys. 275 (2014) 273–293.
[12] H. Hajibeygi, R. Deb, P. Jenny, Multiscale finite volume method for non-conformal coarse grids arising from faulted porous media, in: Proceedings of SPE Reservoir Simulation Symposium, 2011.
[13] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Monotone multiscale finite volume method, Comput. Geosci. (2015) 1–16.
[14] H. Hajibeygi, P. Jenny, Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media, J. Comput. Phys. 228 (2009) 5129–5147.
[15] H. Zhou, H.A. Tchelepi, Operator based multiscale method for compressible flow, SPE J. 13 (2008) 267–273.
[16] P. Jenny, I. Lunati, Modeling complex wells with the multi-scale finite volume method, J. Comput. Phys. 228 (2009) 687–702.
[17] S.H. Lee, H. Zhou, H. Techelpi, Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations, J. Comput. Phys. 228 (2009) 9036–9058.
[18] O. Møyner, K.-A. Lie, The multiscale finite-volume method on stratigraphic grids, SPE J. 19 (2014) 816–831.
[19] H. Hajibeygi, S.H. Lee, I. Lunati, Accurate and efficient simulation of multiphase flow in a heterogeneous reservoir by using error estimate and control in the multiscale finite-volume framework, SPE J. 17 (2012) 1071–1083.
[20] H. Hajibeygi, G. Bonfigli, M.A. Hesse, P. Jenny, Iterative multiscale finite-volume method, J. Comput. Phys. 227 (2008) 8604–8621.
[21] H. Zhou, H.A. Tchelepi, Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, SPE J. 17 (2012) 523–539.
[22] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Algebraic multiscale linear solver for heterogeneous elliptic problems, J. Comput. Phys. 259 (2014) 284–303.
[23] M. Tene, Y. Wang, H. Hajibeygi, Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media, J. Comput. Phys. 300 (2015) 679–694.
[24] U. Trottenberg, C.W. Oosterlee, A. Schueller, Multigrid, Elsevier Academic Press, 2001.
[25] 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.
[26] H. Hajibeygi, H.A. Tchelepi, Compositional multiscale finite-volume formulation, SPE J. 19 (2014) 316–326.
[27] O. Møyner, K.-A. Lie, The multiscale finite volume method on unstructured grids, in: Proceedings of SPE Reservoir Simulation Symposium, 2014.
[28] M. Cusini, A. Lukyanov, J. Natvig, H. Hajibeygi, Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media, J. Comput. Phys. 299 (2015) 472–486.
[29] G. Barenblatt, Y. Zheltov, I. Kochina, Basic concepts in the theory of seepage of homogeneous fluids in fissurized rocks, J. Appl. Math. Mech. 5 (1983) 1286–1303.
[30] J. Warren, P. Root, The behavior of naturally fractured reservoirs, SPE J. 3 (1963) 245–255.
[31] H. Kazemi, Pressure transient analysis of naturally fractured reservoirs with uniform fracture distribution, SPE J. 9 (1969) 451–462.
[32] L.K. Thomas, T.N. Dixon, R.G. Pierson, Fractured reservoir simulation, SPE J. 23 (1983) 42–54.

[33] R. Baca, R. Arnett, D. Langford, Modeling fluid flow in fractured porous rock masses by finite element techniques, Int. J. Numer. Methods Fluids 4 (1984) 337–348.

[34] S.H. Lee, M.F. Lough, C.L. Jensen, Hierarchical modeling of flow in naturally fractured formations with multiple length scales, Water Resour. Res. 37 (2001) 443–455.

[35] S.H. Lee, C.L. Jensen, M.F. Lough, Efficient finite-difference model for flow in a reservoir with multiple length-scale fractures, SPE J. 3 (2000) 268–275.

[36] 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.

[37] J.R. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, S.K. Khataniar, Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs, SPE J. 16 (2009) 880–888.

[38] A.F. Gulbransen, V.L. Hauge, K.-A. Lie, A multiscale mixed finite element method for vuggy and naturally fractured reservoirs, SPE J. 15 (2010) 395–403.

[39] T. Barkve, A. Firoozabadi, Analysis of reinfiltration in fractured porous media, in: Proceedings of SPE Annual Technical Conference and Exhibition, 1992.

[40] J. Noorishad, M. Mehran, An upstream finite element method for solution of transient transport equation in fractured porous media, Water Resour. Res. 3 (1982) 588–596.

[41] M. Karimi-Fard, A. Firoozabadi, Numerical simulation of water injection in 2D fractured media using discrete-fracture model, in: Proceedings of SPE Annual Technical Conference and Exhibition, 2001.

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

[43] A. Moinfar, A. Varavei, K. Sepehrnoori, R.T. Johns, Development of a coupled dual continuum and discrete fracture model for the simulation of unconventional reservoirs, in: Proceedings of SPE Reservoir Simulation Symposium, 2013.

[44] 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.

[45] H. Hajibeygi, Iterative multiscale finite-volume method for multiphase flow in porous media with complex physics, PhD thesis, ETH Zurich, Switzerland, 2011.

[46] A. Moinfar, K. Sepehrnoori, R.T. Johns, A. Varavei, Coupled geomechanics and flow simulation for an embedded discrete fracture model, in: Proceedings of SPE Reservoir Simulation Symposium, 2013.

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

[48] I.S. Ligaarden, M. Krokiewski, K.-A. Lie, D.W. Schmid, M. Pal, On the Stokes–Brinkman equations for modeling flow in carbonate reservoirs, in: Proceedings of ECMOR XII—12th European Conference on the Mathematics of Oil Recovery, 2010.

[49] T.H. Sandve, I. Berre, E. Keilegavlen, J.M. Nordbotten, Multiscale simulation of flow and heat transport in fractured geothermal reservoirs: inexact solvers and improved transport upscaling, in: Proceedings of Thirty-Eighth Workshop on Geothermal Reservoir Engineering, Stanford University, USA, 2013.

[50] M. Tene, M.S. Al Kobaisi, H. Hajibeygi, Algebraic multiscale solver for flow in heterogeneous fractured porous media, in: Proceedings of SPE Reservoir Simulation Symposium, 2015.

[51] M. Tene, M.S. Al Kobaisi, H. Hajibeygi, Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (F-AMS), J. Comput. Phys. (2015), submitted for publication.

[52] O. Møyner, K.-A. Lie, A multiscale method based on restriction-smoothed basis functions suitable for general grids in high contrast media, in: Proceedings of SPE Reservoir Simulation Symposium, 2015.

[53] O. Møyner, K.-A. Lie, A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids, J. Comput. Phys. 304 (2015) 46–71.

[54] O. Møyner, K.-A. Lie, A multiscale restriction-smoothed basis method for compressible black-oil models, SPE J. (2016), http://dx.doi.org/10.2118/173265-PA, in press.

[55] S.T. Hilden, O. Møyner, K.-A. Lie, K. Bao, Multiscale simulation of polymer flooding with shear effects, Transp. Porous Media (2016), http://dx.doi.org/10.1007/s11242-016-0682-2, in press.

[56] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, SPE Reserv. Eval. Eng. 4 (2001) 308–317.

[57] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1998) 359–392.

[58] Y. Wang, H. Hajibeygi, H.A. Tchelepi, Algebraic multiscale solver for flow in heterogeneous porous media, J. Comput. Phys. 259 (2014) 284–303.

[59] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, USA, 2003.

[60] The MATLAB Reservoir Simulation Toolbox (MRST), version 2015a, SINTEF Applied Mathematics, May 2015.

[61] K.-A. Lie, An Introduction to Reservoir Simulation Using MATLAB: User Guide for the Matlab Reservoir Simulation Toolbox (MRST), 2nd edition, SINTEF ICT, Dec. 2015, http://www.sintef.no/Projectweb/MRST/publications.

[62] L.S.K. Fung, A.H. Dogru, Distributed unstructured grid infrastructure for complex reservoir simulation, in: Proceedings of Europec/EAGE Conference and Exhibition, 2008.

[63] A. Kozlova, Z. Li, J.R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, S.H. Lee, A real-field multiscale black-oil reservoir simulator, in: Proceedings of SPE Reservoir Simulation Symposium, 2015.

[64] P. Vanek, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing 56 (1996) 179–196.

[65] P. Vanek, Acceleration of convergence of a two-level algorithm by smoothing transfer operator, Appl. Math. 37 (1992) 265–274.

[66] H. Guillard, P. Vanek, An aggregation multigrid solver for convection–diffusion problems on unstructured meshes, Technical report, Center for Computational Mathematics, University of Colorado at Boulder, 1998.

[67] K.-A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H.M. Nilsen, B. Skaflestad, Open-source MATLAB implementation of consistent discretisations on complex grids, Comput. Geosci. 16 (2012) 297–322.

[68] S. Krogstad, K.-A. Lie, O. Møyner, H.M. Nilsen, X. Raynaud, B. Skaflestad, MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems, in: Proceedings of SPE Reservoir Simulation Symposium, 2015.

[69] K. Bisdom, B.D.M. Gauthier, G. Bertotti, N.J. Hardebol, Calibrating discrete fracture-network models with a carbonate three-dimensional outcrop fracture network: implications for naturally fractured reservoir modeling, Am. Assoc. Pet. Geol. Bull. 24 (2014) 1351–1376.

[70] D.W. Pollock, Semianalytical computation of path lines for finite-difference models, Ground Water 26 (1988) 743–750.

Paper XI

Fully Implicit Simulation of Polymer Flooding with MRST

Kai Bao, Knut-Andreas Lie, Olav Møyner and Ming Liu

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601880

Paper XII

Successful Application of Multiscale Methods in a Real Reservoir Simulator Environment

Knut-Andreas Lie, Olav Møyner, Jostein Roald Natvig, Antonina Kozlova, Kyrre Bratvedt, Shingo Watanabe and Zhouyi Li

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601893

Paper XIII

Nonlinear Solver for Three-phase Transport Problems Based on Approximate Trust Regions

Olav Møyner

In proceedings of the 15th European Conference on the Mathematics of Oil Recovery, 2016, Amsterdam, Netherlands
DOI: 10.3997/2214-4609.201601899

Paper XIV

A Multiscale Restriction-Smoothed Basis Method for Compositional Models

Olav Møyner and Hamdi Tchelepi

Accepted for the proceedings of the SPE Reservoir Simulation Conference 2017, Woodlands, Texas, USA

Is not included due to copyright